



PhD-FSTC-2016-25  
The Faculty of Sciences, Technology and Communication

## DISSERTATION

Defense held on 30/06/2016 in Luxembourg

to obtain the degree of

## DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG EN INFORMATIQUE

by

**Jean-Louis HUYNEN**

Born on 22 December 1983 in Verdun (France)

## SOCIO-TECHNICAL ASPECTS OF SECURITY ANALYSIS

### **Dissertation defence committee**

Dr Ivan Flechais  
*Professor, University of Oxford*

Dr Dieter Gollmann  
*Professor, Hamburg University of Technology*

Dr Vincent Koenig, vice-chairman  
*Université du Luxembourg*

Dr Gabriele Lenzini, chairman  
*Université du Luxembourg*

Dr Peter Y.A. Ryan  
*Professor, Université du Luxembourg*



---

# Abstract

This thesis seeks to establish a semi-automatic methodology for security analysis when users are considered part of the system. The thesis explores this challenge, which we refer to as ‘socio-technical security analysis’. We consider that a socio-technical vulnerability is the conjunction of a human behaviour, the factors that foster the occurrence of this behaviour, and a system. Therefore, the aim of the thesis is to investigate which human-related factors should be considered in system security, and how to incorporate these identified factors into an analysis framework.

Finding a way to systematically detect, in a system, the socio-technical vulnerabilities that can stem from insecure human behaviours, along with the factors that influence users into engaging in these behaviours is a long journey that we can summarise in three research questions:

1. *How can we detect a socio-technical vulnerability in a system?*
2. *How can we identify in the interactions between a system and its users, the human behaviours that can harm this system’s security?*
3. *How can we identify the factors that foster human behaviours that are harmful to a system’s security?*

A review of works that aim at bringing social sciences findings into security analysis reveals that there is no unified way to do it. Identifying the points where users can harm a system’s security, and clarifying what factors can foster an insecure behaviour is a complex matter. Hypotheses can arise about the usability of the system, aspects pertaining to the user or the organisational context but there is no way to find and test them all. Further, there is currently no way to systematically integrate the results regarding hypotheses we tested in a security analysis. Thus, we identify two objectives related to these methodological challenges that this thesis aims at fulfilling in its contributions:

1. *What form should a framework that intends to identify harmful behaviours for security, and to investigate the factors that foster their occurrence take?*
2. *What form should a semi-automatic, or tool-assisted methodology for the security analysis of socio-technical systems take?*

The thesis provides partial answers to the questions. First it defines a methodological framework called STEAL that provides a common ground for an interdisciplinary approach to security analysis. STEAL supports the interaction between computer scientists and social scientists by providing a common reference model to describe a system with its human and non-human components, potential attacks and defences, and the surrounding context. We validate STEAL in a two experimental studies, showing the role of the context and graphical cues in Wi-Fi networks’ security.

Then the thesis complements STEAL with a Root Cause Analysis (RCA) methodology for security inspired from the ones used in safety. This methodology, called S·CREAM aims at being more systematic than the research methods that can be used with STEAL (surveys for instance) and at providing reusable findings for analysing security. To do so, S·CREAM provides a retrospective analysis to identify the factors that can explain the success of past attacks and a methodology to compile these factors in a form that allows for the consideration of their potential effects on a system’s security, given an attacker Threat Model. The thesis also illustrates how we developed a tool—the *S·CREAM assistant*—that supports the methodology with an extensible knowledge base and computer-supported reasoning.

---

## Acknowledgements

Firstly, I would like to thank the members of my supervisory committee. Gabriele Lenzini for his daily supervision, his wise advices, and for helping me clarifying and expressing my ideas. Vincent Koenig, for his support, his insights on user studies, psychology and safety, and for the tremendous amount of feedback he gave me on my dissertation. Peter Ryan, for welcoming me in his team, for the constructive feedback he gave me on my work, and his invaluable insights on other research works.

I address a special thanks to Rosario Giustolisi, Ana Ferreira, and Salvador Rivas. I had pleasant time working with them and the completion of parts of this thesis work would not have been possible without their help.

My special thanks also go to Eric François, Sophie Doublet, and Katja Weinerth for the support they provided. It made some parts of my work a lot easier.

I also wish to thank all the members of the APSIA and the COSA research teams, as well as all the other people from the university of Luxembourg and SnT that I met in the different reading groups, seminars, conferences, and lectures. I'll remember the lively discussions and the good atmosphere.

Moreover, I would like to thank my Ph.D. examiners Dieter Gollmann and Ivan Flechais who accepted to take the time to read and to evaluate my dissertation. They offered me constructive feedback that helped me to improve this manuscript.

I wish to acknowledge the support of the Fond National de la Recherche (FNR) through the Socio-Technical Analysis of Security and Trust (STAST) project I2R-APS-PFN-11STAS.

Finally, I'm very grateful to my family, my family in law, and my friends for their support, understanding, and patience during the time of my Ph.D. studies. Nothing would have been possible without them.

---

*à Marine, Gaston et Émile,*



# Contents

Abstract . . . . .	iii
Acknowledgements . . . . .	iv
List of Tables . . . . .	xi
List of Figures . . . . .	xiv
<b>I Background, Objectives, and Research Questions</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 The human factor of security . . . . .	3
1.2 Contributions . . . . .	6
1.3 Thesis structure . . . . .	8
1.3.1 Publications . . . . .	9
<b>2 Research background</b>	<b>11</b>
2.1 Socio-Technical Attacks . . . . .	11
2.1.1 Socio-Technical Attack (STA) without <i>malicious intent</i> . . . . .	13
2.1.2 STA with extraneous <i>malicious intent</i> . . . . .	15
2.1.3 STA with user's <i>malicious intent</i> . . . . .	17
2.2 Socio-Technical Security Analysis . . . . .	18
2.3 Different audiences with different goals . . . . .	20
2.4 Inspiration from the safety field . . . . .	21
2.5 Research Questions and Objectives . . . . .	22
<b>II STEAL, a framework for socio-technical security analysis, and applications</b>	<b>25</b>
<b>3 STEAL: a framework for socio-technical security analysis</b>	<b>27</b>
3.1 Introduction . . . . .	28
3.2 The STEAL framework . . . . .	29
3.2.1 A socio-technical security conceptual framework . . . . .	29
3.2.2 Illustrating the use of STEAL: Wi-Fi networks selection . . . . .	34
3.2.3 Summary . . . . .	37
3.3 STEAL validation: social focus . . . . .	37
3.3.1 Introducing the study . . . . .	38
3.3.2 Use Case Scenario . . . . .	39
3.3.3 The Survey . . . . .	40

3.3.4	Context . . . . .	41
3.3.5	Security Discussion . . . . .	45
3.3.6	Conclusion . . . . .	45
3.4	STEAL validation: technical focus . . . . .	46
3.4.1	Introduction . . . . .	47
3.4.2	Study . . . . .	48
3.4.3	Related Work . . . . .	50
3.4.4	Conclusion . . . . .	50
3.5	Discussion . . . . .	51
3.5.1	Reference Model . . . . .	51
3.5.2	Support to social and computer sciences methods for security analysis . . . . .	51
3.6	Conclusion . . . . .	52
<b>4</b>	<b>Applying STEAL for the Socio-Technical security analysis of Wi-Fi hotspots</b>	<b>55</b>
4.1	Introduction . . . . .	56
4.2	Raising Research Questions on critical actions . . . . .	57
4.2.1	Methods . . . . .	57
4.2.2	Use cases . . . . .	59
4.2.3	Informal Technical Security Analysis . . . . .	63
4.2.4	Discussion . . . . .	65
4.2.5	Conclusion . . . . .	67
4.3	Investigating Research Questions . . . . .	67
4.3.1	The influence of trust in the selection of Wi-Fi networks. . . . .	68
4.3.2	The influence of graphical cues in the selection of Wi-Fi net- works. . . . .	73
4.4	Conclusion . . . . .	83
<b>III</b>	<b>S-CREAM, a Root Cause Analysis methodology for socio-technical security</b>	<b>85</b>
<b>5</b>	<b>Beyond STEAL: building an RCA for security</b>	<b>87</b>
5.1	Introduction . . . . .	87
5.2	RCA in Safety and Security: Differences and Challenges . . . . .	89
5.2.1	RCA in Safety . . . . .	90
5.2.2	From RCA to prediction . . . . .	91
5.2.3	RCA in Safety and Security: Differences . . . . .	92
5.2.4	Towards an RCA for Socio-Technical Security: Challenges . . . . .	98
5.3	Conclusion . . . . .	100
<b>6</b>	<b>S-CREAM: an RCA for socio-technical security</b>	<b>101</b>
6.1	Introduction . . . . .	102
6.2	S-CREAM's process and concepts . . . . .	103
6.2.1	Overview . . . . .	103
6.2.2	The failure of the generic RCA process . . . . .	103



6.2.3	S-CREAM's overall process . . . . .	104
6.2.4	First step: Data collection, and investigations . . . . .	105
6.2.5	Second step: Retrospective Analysis . . . . .	107
6.2.6	Third step: Generalisation . . . . .	108
6.2.7	Fourth step: Security Analysis . . . . .	110
6.2.8	Controlling socio-technical vulnerabilities . . . . .	112
6.3	S-CREAM's implementation . . . . .	112
6.3.1	Implementation Choices . . . . .	113
6.3.2	Implementing the <i>Data Collection, and Investigations</i> step of S-CREAM . . . . .	114
6.3.3	Implementing the <i>Retrospective Analysis</i> of S-CREAM . . . . .	115
6.3.4	Implementing the <i>Generalisation</i> step of S-CREAM . . . . .	116
6.3.5	Bootstrapping S-CREAM's catalogue of Attack Modes . . . . .	117
6.3.6	Implementing the <i>Security Analysis</i> of S-CREAM . . . . .	121
6.4	S-CREAM's companion tool: <i>S-CREAM assistant</i> . . . . .	122
6.4.1	Main functions . . . . .	122
6.4.2	Additional features . . . . .	126
6.4.3	Technical implementation . . . . .	126
6.5	Conclusion . . . . .	127
<b>7</b>	<b>Applying S-CREAM for socio-technical security analysis</b>	<b>129</b>
7.1	Introduction . . . . .	130
7.2	Authentication of identities with TLS certificates in web browsers . . . . .	131
7.2.1	Description . . . . .	131
7.2.2	Threat Model . . . . .	131
7.2.3	Semi-automatic Security Analysis . . . . .	131
7.2.4	Analyst-driven Security Analysis . . . . .	132
7.2.5	Discussion on the results and the possible remediations . . . . .	132
7.3	Hotspot use cases . . . . .	133
7.3.1	Description . . . . .	133
7.3.2	Threat Model . . . . .	134
7.3.3	Semi-automatic Security Analysis . . . . .	135
7.3.4	Analyst-driven Security Analysis . . . . .	135
7.3.5	Discussion on the results and the possible remediations . . . . .	136
7.4	YubiKeys use case . . . . .	137
7.4.1	Description . . . . .	138
7.4.2	Threat Model . . . . .	138
7.4.3	Semi-automatic Security Analysis . . . . .	138
7.4.4	Analyst-driven Security Analysis . . . . .	138
7.4.5	Discussion on the results and the possible remediations . . . . .	139
7.5	Conclusion . . . . .	140
<b>8</b>	<b>Review of how S-CREAM meets our challenges</b>	<b>143</b>
8.1	Introduction . . . . .	143
8.2	How S-CREAM meets our challenges . . . . .	144
8.2.1	Challenge $C_4$ : Matching patterns of known attacks . . . . .	144
8.2.2	Challenge $C_3$ : Creating reusable knowledges . . . . .	146

8.2.3	Challenge $C_2$ : Investigating Attacks . . . . .	147
8.2.4	Challenge $C_1$ : Addressing the lack of knowledge and structured data to support the analysis . . . . .	149
8.2.5	Challenge $C_5$ : Being flexible . . . . .	150
8.3	Conclusion . . . . .	150
<b>IV</b>	<b>Discussion and Conclusion of the thesis work</b>	<b>153</b>
<b>9</b>	<b>Conclusion</b>	<b>155</b>
9.1	Summary . . . . .	155
9.2	Contributions . . . . .	156
9.3	Discussion . . . . .	157
9.4	Directions for future works . . . . .	160
<b>A</b>	<b>Complementary social analysis of the TLS validation use case</b>	<b>163</b>
<b>B</b>	<b>Questionnaire used for testing the influence of graphical cues in the selection of Wi-Fi networks</b>	<b>167</b>
<b>C</b>	<b>Pilot study performed to prepare the questionnaire presented in Appendix B</b>	<b>179</b>
C.1	The vignettes . . . . .	179
C.2	The questionnaire . . . . .	179
C.3	Choosing the best vignette to convey the scenario's meaning . . . . .	182
<b>D</b>	<b>Questions used in the study on the influence of the context and trust on Wi-Fi network selection</b>	<b>187</b>
D.1	Questions common to both conditions . . . . .	187
D.2	Trust condition . . . . .	188
D.3	Context condition . . . . .	188
	<b>Bibliography</b>	<b>191</b>

# List of Tables

1.1	This table details how parts of my previously published papers have been used throughout this thesis. To sum up its content, Part II reuses a lot of materials, whereas Parts I, III, and IV contain mostly original (and individual) content, which we intend to publish in the near future. . . . .	10
3.1	Existing/nonexistent wireless names and their grouping in relation to security and context. Security: (G1-existing; G2-nonexistent; G3-nonexistent and related to security; G4-nonexistent and not related to security). Context: (L1-existing and expected in the context; L2-existing and not expected in the context; L3-nonexistent and expected in the context; L4-nonexistent and not expected in the context). . . .	39
3.2	Sociodemographics for the population of the survey for the <i>context</i> condition, and to whole survey . . . . .	41
3.3	Statistical significance for the differences between general preferences and the contexts (in this case, there is no statistical significance for the context “city center”). . . . .	41
3.4	Statistical significance for the differences between general preferences and the context of the University. . . . .	43
3.5	Statistical significance for the differences between general preferences and the context for: (a) Shopping Mall, and (b) the Hospital. . . . .	44
3.6	Most common reasons for general preferences and each context. . . . .	44
4.1	Socio-technical security analysis of the classic pay as you go captive portal . . . . .	64
4.2	Socio-technical security analysis of an automatic roaming to a Hotspot 2.0 through an ANDSF policy . . . . .	65
4.3	Sociodemographics for the population of the survey for the <i>trust</i> condition, and to whole survey . . . . .	69
4.4	Statistical significance for the differences between: (a) general preferences and trust; (b) general preferences and trust but for groups G1-G4 (G1-existing; G2-nonexistent; G3-nonexistent and related to security; G4-nonexistent and not related to security). . . . .	70
4.5	Most common reasons related to general preferences (G) and trust (T) for all choices, choices that change to nonexistent names (CPTUN), or to nonexistent names related to security (CPTSN), and that do not change from general preferences to trust. . . . .	72

4.6	Chosen vignettes to convey the need for ‘Encryption’ or ‘QoS’ and their limitations. . . . .	78
4.7	Sociodemographics profile by scenario. . . . .	79
4.8	Counts and frequencies for the third round of the study. . . . .	80
4.9	Trimmed results of the logistic regression of network selection on password + scenario. (S0-0 reference category) . . . . .	80
4.10	Exponentiated coefficients of the logistic regressions for the main effect of “Encryption” and “QoS” while controlling for password and scenario. LR tests compare models with and without interaction terms. 81	
4.11	Logistic regression results. Tests are performed between the current model and the previous one. AIC is evaluated as well. (* < .05; ** < .01; *** < .001) . . . . .	81
6.1	Table listing the Attack Modes (AMs) corresponding to the <i>Identity Spoofing</i> and the <i>Action Spoofing</i> Socio-Technical Capabilities (STCs). The right column displays the Contributors that the Attack Pattern (AP) allowed us to identify as linked to the STC. Contributors are only appended once to the list of each STC’s Contributors and other components of AMs such as Pre-Requisites (PRs) are not displayed for the sake of space. APs are sorted numerically. . . . .	119
6.2	Justifications for the selections of Contributors for the ‘Wrong identification’ generic antecedent. Specific antecedents inside generic antecedents are not displayed and specific antecedents and generic antecedents are abbreviated as SA and GA. . . . .	120
6.3	Reduced list of <i>AMs</i> for the <b><i>Identity spoofing</i></b> <i>STC</i> . The attacker’s capabilities have to match the set of the <i>AM</i> ’s PRs in order for the attacker to be able to use the AM. For instance (see highlighted row), an attacker has to be in control of his ‘Declared identity’ (he can declare himself as anyone) in a message that is a continuation of a previous interaction between the victim and the source (on any medium) to gain the <b><i>Identity spoofing</i></b> <i>STC</i> by exploiting the ‘Incorrect label’ Contributor. AMs that only require the attacker to be able to send a message are not shown for the sake of space, namely: <i>Bad mental Models</i> , <i>Mislearning</i> , and <i>Multiple Signals</i> . . . . .	121
7.1	Lists of Contributors of the <b><i>Identity spoofing</i></b> <i>STC</i> and of the potential Man In The Middle attack. Shared Contributors are highlighted in blue. . . . .	132
7.2	Potential factors that we identified in the previous works that can potentially foster the insecure behaviours. . . . .	134
7.3	Results of each analyst-driven <i>Security Analysis</i> for each Hotspot use case grouped by insecure behaviours. . . . .	135
7.4	Contributors yielded by the <i>Security Analysis</i> . . . . .	139
8.1	Links between the challenges stated in Chapter 5 and S-CREAM’s steps introduced in Chapter 6. Each challenge depends on its predecessor except $C_5$ , which is the subsidiary challenge of being flexible. .	144

C.1	Vignettes that intend to convey the lack of need for ‘Encryption’ and ‘QoS’.	180
C.2	Vignettes that intend to convey the lack of need for ‘Encryption’, and the need for ‘QoS’.	180
C.3	Vignettes that intend to convey the need for ‘Encryption’, and the lack of need for ‘QoS’.	181
C.4	Vignettes that intend to convey the need for ‘Encryption’ and ‘QoS’.	182



# List of Figures

2.1	This figure shows the overlaps between the different STAs presented. Semantic Attacks are part of Social Engineering. Deliberate Non-Compliance is part of Human Error. Insider Threat overlaps with Social Engineering attacks and Human Errors, and malicious users are only part of Insider Threat related research. The ‘Well Meaning Users’ area comprehends STAs without <i>malicious intent</i> and STAs with extraneous <i>malicious intent</i> . . . . .	13
3.1	STEAL Conceptual Framework. $\nabla$ are possible attacks, and $\ominus$ are possible defences. The attacker can strike in the context, and in every layer between the user and the network. . . . .	30
3.2	Operational guideline to use Technical and Social Analyses in a pipeline.	33
3.3	Reference model for WiFi connection to the Internet. . . . .	34
3.4	Technical Focus: the UML sequence diagram for the WiFi connection to the Internet with an intruder attacking the network. . . . .	36
3.5	The multi-layered security and threat model used in this section. . . .	38
3.6	Selection made for eduroam, Hotel.le_Place_d’Armes, secure_wifi_BelleEtoile and free-wifi_BelleEtoile within the four contexts by all participants of condition 2. . . . .	42
3.7	General preferences in the 4 groups (L1-L4) for all the contexts. . . .	43
3.8	The multi-layered security and threat model used in this section. There is no context. The intruder is a man-in-the-middle represented by the inverted triangle. . . . .	46
3.9	Activity diagram for TLS certificate validation in Chrome . . . . .	49
4.-1	(a) UML diagram depicting user’s interaction when joining a pay-per-use Hotspot. Components are at the top of each line, arrows represent exchanged messages: a plain line is used when a component initiates a message and a dotted line when a component replies to a message. The blue tunnels represent the messages communicated within an encrypted tunnel with HTTPS. (b) UML diagram depicting user’s interaction when Roaming on a Hotspot 2.0 Hotspot. Components are at the top of each line, arrows represent exchanged messages: a plain line is used when a component initiates a message and a dotted line when a component replies to a message. The blue tunnels represent the messages encrypted using WPA2 protocol or communicated within an encrypted tunnel with HTTPS. . . . .	62

4.0	General preferences vs. trust. in condition 1 for each SSID. . . . .	70
4.1	General preferences vs. trust for groups G1-G4. . . . .	71
4.2	Rounds of choices. . . . .	76
4.3	Age and Occupation distribution for Males and Females . . . . .	80
5.1	The process usually used for a Root Cause Analysis. . . . .	90
5.2	Reason’s Swiss cheese model . . . . .	90
6.1	The generic process of an RCA against the challenges introduced in Section 5.2.4. . . . .	103
6.2	S-CREAM process and steps. The figure’s <i>Security Analysis</i> step (see Section 6.2.7) sketches the semi-automatic modality. An expert-driven modality reuses the <i>Data Collection and Investigations</i> and the <i>Retrospective Analysis</i> steps to investigate potential attacks, and then appends its products to the semi-automatic <i>Security Analysis</i> ’s results. Acronyms used in the figure: PR = Pre-Requisites, Co = Contributor, Ef = Effect, STC = Socio-Technical Capability, TM = Threat Model. . . . .	105
6.3	Side-by-side comparison between Cognitive Reliability and Error Analysis Method (CREAM) (left-hand side) and S-CREAM (right-hand side) processes. . . . .	116
6.3	(Previous page) Screen capture of <i>S-CREAM assistant</i> while performing the <i>Retrospective Analysis</i> of the ‘Wrong Object’ Error Mode (EM) observed in CAPEC-195. A green light means that we consider the specific antecedent to be a Contributor or that we expand the generic antecedent. A red light means that we do not consider that the antecedent contributes to the EM. Stop signs show where stop rules are engaged. . . . .	124
6.4	Screen capture of the attacker manager for STCs in <i>S-CREAM assistant</i> . . . . .	125
6.5	<i>S-CREAM Assistant</i> data model. Tables in blue are implemented using js-data and stored in the web-browser’s Local Storage. . . . .	126
A.1	Sequence diagram for the studied attack scenario. We use alternatives this way to emphasise the needed actions to make the attack fail or succeed. The possible use of the Context $C_A$ as a defence is not represented here for the sake of simplicity. . . . .	164
B.1	This is the questionnaire’s landing page that the participant reaches when picking our HIT from the amazon mechanical turk HIT list. . .	168
B.2	First, we ask for the participant’s consent to participate to the study.	169
B.3	Then, we check whether the participant is technically able to participate. . . . .	169
B.4	In this first part of the collection of data, we retrieve socio-demographic information from the participants. . . . .	170



B.5	This page gives instructions to the participant regarding the survey. Furthermore, the participant is randomly assigned to a condition (with password / without a password), in this particular case, the participant is given a password. . . . .	170
B.6	This is the 1st choice out of 5. In this particular case, the 1st choice is a 2nd round. The participant is assigned to a scenario displayed on the top of the page. Once defined, the same scenario is used for all the subsequent choices this participant is asked to make. . . . .	171
B.7	This is the 2nd choice out of 5. In this particular case, the 2nd choice is a 3rd round. . . . .	171
B.8	This is the 3rd choice out of 5. In this particular case, the 3rd choice is a 1st round. . . . .	172
B.9	This is the 4th choice out of 5. In this particular case, the 4th choice is a 3rd round. This 3rd round has been reintroduced randomly to check the consistency of the participant’s answers. . . . .	172
B.10	This is the last choice. In this particular case, the 5th choice is a 4th round. In a 4th round, the networks names appear one after the other, each subsequent addition is delayed by 200ms. . . . .	173
B.10	(previous page) Here, we ask the participants to rate on 4-points likert scales, their understanding of the extent to which they agree that each of the 3 visual cues corroborate in meaning with 4 words related to ‘Encryption’, and ‘QoS’. . . . .	175
B.11	After having investigated the participant’s understanding of the graphical cues, we ask complementary questions about the participant’s attitude and belief regarding the participant use of Wi-Fi networks. . . . .	175
B.12	In this last page of collection of data, we continue our investigation of the participant’s beliefs, and we ask the participant about his connectivity habits. . . . .	176
B.13	Finally, we thank the participant and redirect him to the amazon mechanical turk website. . . . .	177
C.1	This page gives instructions to the participant regarding the survey. .	181
C.2	This page asks the participant to rate, for the described scenario, how much each property is required by the Wi-Fi network. The participant is asked to perform this task for the 12 vignettes. . . . .	183
C.3	This page gathers information regarding the participant’s connectivity habits, and the participant’s beliefs about Wi-Fi networks. . . .	184
D.1	Illustration of the actual presentation of the online questionnaire, when asking for the participant’s preferences regarding network names, in English. This question is presented several times to the participants throughout the questionnaire, with different instructions and likert scale ratings. . . . .	188



# Part I

## Background, Objectives, and Research Questions



*La conscience qui s'émeut ressemble assez à la conscience qui s'endort.*

—Jean-Paul Sartre, 'Esquisse d'une théorie de l'émotion'

# 1

## Introduction

### Contents

---

<b>1.1</b>	<b>The human factor of security . . . . .</b>	<b>3</b>
<b>1.2</b>	<b>Contributions . . . . .</b>	<b>6</b>
<b>1.3</b>	<b>Thesis structure . . . . .</b>	<b>8</b>
1.3.1	Publications . . . . .	9

---

### 1.1 The human factor of security

Even 'secure' systems can turn out to be vulnerable when attackers target not the technical system and its security mechanisms but the people interacting with it. In such situations, security is not a purely technical property but rather a socio-technical quality stemming from factors such as people's interactions with the technology and the underlying cognitive and psychological factors.

While security experts are just starting to explore this new field, hackers have already mastered the art. Verizon's 2015 Data Breach Investigation Report [26] assessed nearly 80,000 security incidents that occurred across 61 countries in 2015. It states that: 'the common denominator across the top four patterns —accounting for nearly 90% of all incidents— is people.'

If a large majority of attacks use people to trespass into computer systems, it is simply because it works. Phishing, for instance, is often the first step of an attack and has been proven to be highly effective. A phishing campaign of 10 emails has a 90% chance to yield at least one victim, and across the campaigns that Verizon analysed, the median time for the first victim's click was less than two minutes. Worse, attackers are getting better at it as the overall effectiveness of phishing campaigns jumped from 10%–20% in 2014 to 23% in 2015.

Such dramatic figures make the study of the role of the human component in security appealing but, alas, it is a complex matter. Even without attempting to model the human mind and its intricate processes that drive people's choices, the problem seems hard to grasp. A few general behavioural and cognitive principles have been identified and can be applied in the field of security (e.g., see [55, 172, 9, 3, 183, 50, 165]), but, when people face several alternatives, it is close to impossible to identify why one alternative is chosen over another. Furthermore, the decision-taking process is not deterministic, and as Gilovich puts it: 'The tiniest little change in circumstance can have big impacts on people's behaviour.' [84].

Thus, attempting to model the processes that drive human behaviour in security seems intractable, and we are left with the study of what surrounds this process: its inputs and outputs. In other words, we study the factors that potentially influence the human behaviours that have consequences on a system's security without attempting to model humans.

These factors are very diverse and constitute the context, or as mentioned earlier, the 'circumstances' of a user action. Some factors pertain to the user, for instance, the user's computer literacy, the user's language and culture, or the state of mind or persona that a person has when he is interacting with the system. The location in terms of space and time as well as the interactions with the other actors are additional factors. Besides, the system itself, its affordance (the way in which the system presents the available actions), the instructions it provides, and the signs and symbols it uses to do so—all play a role in the user's behaviour.

Furthermore, these factors can have a different effect on the user when combined with each other. For instance, the user's understanding of a concept depends on the context in which the concept is used [25]. For instance, the users' culture can also function as an interfering factor, and in addition to words, concepts and stereotypes as well as symbols and signs can have different meanings and represent different things depending on who sees them [136].

These influencing factors can potentially nudge people into behaviours harmful to the system's security or facilitate attacks launched by a malicious actor. For instance, a user can misunderstand how to operate the system and commit a mistake; a user can bypass a system's security mechanism because he deems that the mechanism impinges on the system's primary function to an extent that it decreases performance in a way that is simply unacceptable; a user can be gullible and concede valuable information to a social-engineer; or an attacker can exploit a loophole in the user-context-system interactions to push a user to execute an action on the system on his behalf. In all of these cases, we consider that a successful socio-technical attack on a system is evidence that the system was doomed to fail, and that we should not (necessarily) blame the user. It is likely that the blame lies largely with the designers for failing to take properly into account the human-related factors.

**Vulnerabilities in socio-technical analysis.** This relation between insecure human behaviours and the factors that influence these behaviours has consequences on what we consider as vulnerability in a system. For instance, a user can send a recipient his private key, jeopardising the private key's confidentiality. But in itself, it is not a vulnerability; it is an insecure behaviour that has consequences on secu-

ity. A (socio-technical) vulnerability arises from the fact that we call keys ‘private’ and ‘public’, and that it can cause confusion in the user’s mind, which can result in a mistake or an insecure behaviour that harms security. Therefore, a socio-technical vulnerability is the conjunction of a behaviour harmful for security, of a factor that fosters this behaviour, and the system whose security is endangered.

**Exploiting these socio-technical vulnerabilities.** Confidence tricksters have known for long that there are factors facilitating scams as well as about the existence of new technology, which not only change the tools of the trade but also how victims are approached, selected, and robbed. For instance, the Nigerian 419 advance-fee scams are nothing new and find their roots in an older scam called the ‘Spanish prisoner’ [170] dating back to the 16th century. In the original ‘Spanish prisoner’ scam, the confidence trickster tells the victim that the victim’s help is needed to free a wealthy person imprisoned in Spain. The prisoner promises a reward for his release and sometimes even the hand of his daughter. The catch is that the victim has to provide funds before receiving any reward in order to free the fictitious prisoner. The novelty of the contemporary ‘Spanish prisoner’, like the Nigerian 419 scams, is that the Internet gives scammers access to a huge pool of potential victims at a negligible cost. Hence, whereas in the past con-artists had access to few potential victims and had to find the right incentive in order to extort money, nowadays, scammers only have to send the same generic e-mail en masse to run their attack.

However, scammers performing 419 scams actually select people to be scammed on the basis on their predispositions or facilitating personal factors. Indeed, for an attacker with limited resources, being able to reach thousands of people is actually both a blessing and a curse because people swallowing the initial bait may not follow through the rest of the profit-yielding scam. To avoid wasted efforts, scammers craft their hooks and baits carefully. It is likely that the typical 419 scam emails are poorly written and carry all the stereotypes of this scam to repel savvy people and cherry-pick the most gullible victims, in order to maximise profits [88]. Attackers know from experience which factors make people worth their time, and therefore, they have strategies to focus only on these people. Notably, it is no different when it comes to attacking a corporate network.

**Identifying and controlling these socio-technical vulnerabilities.** We believe that identifying the factors that facilitate socio-technical attacks is at the heart of the analysis of socio-technical systems, and that these factors constitute the entry-points or the socio-technical vulnerabilities that need to be identified and controlled.

In security, there is an asymmetry between attack and defence, and socio-technical attacks are no exception to this observed norm. Attackers have an advantage: finding one vulnerability is easier than protecting the whole system, which requires finding and fixing all the vulnerabilities. When an attacker is only interested in getting the right fit for an attack, he ‘only’ has to find which factors to exploit given the system, its users and the surrounding context. A security researcher has a far more difficult task: identifying the potential vulnerable combinations of people, context, and system features that could harm the system’s security. However, this duality offers the security researchers an interesting per-

spective: they can take the intruder’s viewpoint, plan and assess socio-technical attacks, and thereafter, change hats and take the security engineering side, this time trying to patch the discovered vulnerabilities.

Besides, there is one underexploited aspect that we think could shift the balance in favor of security researchers to an extent: they have access to the data. Indeed, security researchers can keep track of attacks, launch investigations to gather additional details about attacks, study the socio-technical vulnerabilities that made these attacks successful, and use this knowledge to secure their systems. But while security researchers engage in threat intelligence and sharing, we believe that some important data related to the factors exploited by socio-technical attacks are not identified or marked for collection, and that this undermines the efforts of the security researchers.

Either way, we believe that both approaches of testing hypotheses about potential factors, and learning from past socio-technical attacks are valuable to analysing a socio-technical security and this thesis intends to demonstrate that.

## 1.2 Contributions

Making a system effectively secure is a complex matter because the systems used by people are inherently socio-technical. A system can be technically secure but can still fail to provide effective security because its users undermine its security measures. For example, in hospitals, access controls will be violated unless designed to fit the nomadic, interrupted, and cooperative nature of the medical work. Thus, providing an effectively secure system for such a context of work calls for diverse strategies, solutions, and skill sets.

This diversity is an obstacle when it comes to studying the security of socio-technical systems. For instance, the effective security of a security mechanism with a graphical user interface will require the use of both computer sciences and social sciences methods to be tested. The security protocol that the security mechanism implements may have to be studied through the use of formal methods, and the graphical user interface’s usability may require to be tested through surveys or in-lab experiments.

Having different scientists with their own methods, vocabulary, and priorities working on different aspects of a system makes the security analysis complex and can harm the validity of the analysis. For instance, a computer scientist may think that it is appropriate to ask users to use passwords that are 20 characters long at minimum, and social scientist may imagine that providing immediate feedback for a failed password attempt is the right thing to do. However, both are problematic from an effective security point of view [10].

This thesis provides two main contributions. **The first contribution is a framework that aims at testing hypotheses about factors that can constitute socio-technical vulnerabilities. The second contribution is a methodology that allows a security researcher to identify the factors that make an attack successful, and to identify socio-technical vulnerabilities that exploit these factors in a system.** To do so, we bring together methodolo-



gies and tools from multiple fields of research towards a common goal: studying and preventing socio-technical attacks to make socio-technical systems effectively secure.

Our first contribution is a framework called the Socio-TEchnical Attack AnaLysis (STEAL) framework that provides a common ground for an interdisciplinary approach towards security analysis. STEAL supports the interaction between computer scientists and social scientists by providing a common reference model to describe a system, potential attacks and defences, and the surrounding context in terms of layered security ceremonies. This reference model allows for different researchers with different backgrounds, skills, and methods to work on the same system. Computer scientists use formal methods to tackle the technical aspects of security and social scientists perform user studies and other experiments to investigate factors pertaining to the user, the user interface or the context that could undermine the system's security.

Our second contribution is a methodology called the Cognitive Reliability and Error Analysis Method for Socio-Technical security (S-CREAM), which is inspired from the Root Cause Analysis (RCA) techniques found in the safety field. S-CREAM supports security analysts in different tasks related to incident response and security analysis. Indeed, S-CREAM offers guidance to collect data relevant to a security incident and as well as means to identify factors that contributed to the success of an attack. Furthermore, S-CREAM allows the analyst to compile his findings into a knowledge catalogue, which can be later used to test a system for socio-technical vulnerabilities. This set of feature allows S-CREAM to tackle the security analysis of a wide range of systems and systematically apply lessons learnt from past attacks.

**The contributions of this thesis are therefore summarised as follows:**

- **STEAL, a framework for socio-technical security analysis (in Part II),**
- **S-CREAM, a Root Cause Analysis methodology for socio-technical security (in Part III).**

**Targets audiences and key objectives.** The STEAL framework and the S-CREAM methodology have different primary audiences and objectives. STEAL intends to provide security researchers of different backgrounds with a common ground where to formulate and answer research questions about the effectiveness of a security mechanism. It intends to be used for close and thorough inspections of security mechanisms through the use of scientific methods and to produce results whose quality is aligned with the researcher's scientific rigour.

S-CREAM has a more pragmatic approach for producing results. Indeed S-CREAM does not hassle security practitioners with a fully fledged scientific process, and instead, builds upon an existing RCA method called Cognitive Reliability and Error Analysis Method (CREAM) to provide reasonable evidence on the influence of human-related factors on a system's security. S-CREAM intends to match security practitioners' needs for a tool that allows for a quick overview of the human-related factors that lead or can lead to a security compromise.

## 1.3 Thesis structure

The thesis is structured in nine chapters.

In Chapter 2, we start by examining a non-exhaustive list of research works that, in the computer security landscape, have tried to improve computer security methods by incorporating insights from social sciences. These works call for the development of a framework that allows for a better collaboration between social and computer scientists and forward a methodology to streamline the use of the findings from social sciences and investigations of past attacks in security analysis.

In Chapter 3, we introduce the STEAL framework, a framework to study socio-technical security. The main goal of STEAL is to provide computer scientists and social scientists with a common ground and a similar terminology to allow for a security analysis that draws from technical, social, and contextual elements. In the first part of this chapter, we present the framework together with its reference model and associated methodologies. Thereafter, we validate its relevance for social scientists through a study on the role of context and names in the Wi-Fi selection process. Eventually, we validate its relevance for computer scientists by a study of the validation of identities using TLS certificates in web-browsers.

In Chapter 4, we demonstrate a user-centric approach to socio-technical security analysis by studying Wi-Fi hotspots' most salient security ceremonies and compiling potential attacks and research questions about the reasons behind their success. Thereafter, we investigate two of the previously identified research questions about the Wi-Fi selection process: the first about the role of trust, the second about the role of graphical cues.

In Chapter 5, we discuss the requirements for a Root Cause Analysis (RCA) for computer security. We start by introducing RCA techniques, their origin, and use in the safety field. Then, we identify the main differences between the safety and security fields that lead to a list of challenges to be considered when building an RCA for security.

In Chapter 6, we present the Cognitive Reliability and Error Analysis Method for Socio-Technical security (S-CREAM), an RCA technique for security. S-CREAM embraces the analysis of root causes, and is hence, a tool capable of retrospective as well as prospective socio-technical analyses. First, we introduce the technique's process and concepts, and then, we elaborate upon the implementation of S-CREAM along with a companion tool: *S-CREAM assistant*.

In Chapter 7, we use S-CREAM on several use cases to stress its relevance to security. The use cases are as follows: the validation of TLS certificates in web browsers, Wi-Fi hotspots, and Yubikeys security tokens.

In Chapter 8, we discuss how well S-CREAM met the challenges we listed in Chapter 5 and what improvements should be made to the methodology.

In Chapter 9, we sum up the thesis work and contributions, and describe how STEAL and S-CREAM fulfil the objectives as well as how they can benefit the different audiences. Finally, we propose some directions for future works.

### 1.3.1 Publications

Parts of this thesis are revised and updated versions of full texts or parts of works of the following authors<sup>1</sup>:

1. A. Ferreira, JL. Huynen, V. Koenig, G. Lenzini, and S. Rivas. **Do Graphical Cues Effectively Inform Users? - A Socio-Technical Security Study in Accessing Wifi Networks.** In *HCI (22)*, volume 9190 of *Lecture Notes in Computer Science*, pages 323–334. Springer, 2015. [Best Paper Award]
2. A. Ferreira, JL. Huynen, V. Koenig, and G. Lenzini. **In Cyber-Space No One Can Hear You S-CREAM - A Root Cause Analysis for Socio-Technical Security.** In *STM*, volume 9331 of *Lecture Notes in Computer Science*, pages 255–264. Springer, 2015
3. A. Ferreira, JL. Huynen, V. Koenig, and G. Lenzini. **Socio-technical Security Analysis of Wireless Hotspots.** In *HCI (24)*, volume 8533 of *Lecture Notes in Computer Science*, pages 306–317. Springer, 2014
4. A. Ferreira, JL. Huynen, V. Koenig, and G. Lenzini. **A Conceptual Framework to Study Socio-Technical Security.** In *HCI (24)*, volume 8533 of *Lecture Notes in Computer Science*, pages 318–329. Springer, 2014
5. A. Ferreira, JL. Huynen, V. Koenig, G. Lenzini, and S. Rivas. **Socio-Technical Study on the Effect of Trust and Context When Choosing WiFi Names.** In *STM*, volume 8203 of *Lecture Notes in Computer Science*, pages 131–143. Springer, 2013
6. A. Ferreira, R. Giustolisi, JL. Huynen, and G. Lenzini. **On Tools for Socio-Technical Security Analysis.** Grande Region Security and Reliability Day, 2013
7. A. Ferreira, R. Giustolisi, JL. Huynen, V. Koenig, and G. Lenzini. **Studies in Socio-technical Security Analysis: Authentication of Identities with TLS Certificates.** In *TrustCom/ISPA/IUCC*, pages 1553–1558. IEEE Computer Society, 2013

Each chapter recalls in its preamble which papers have been used, and Table 1.1 details what contents of which papers each Chapter uses.

---

<sup>1</sup>Authors are ordered alphabetically.

Table 1.1: This table details how parts of my previously published papers have been used throughout this thesis. To sum up its content, Part II reuses a lot of materials, whereas Parts I, III, and IV contain mostly original (and individual) content, which we intend to publish in the near future.

Part	Chapter	Publications	Use
I	1	All publications	Parts of the introductions
	2	All publications	Parts of the related works
II	3	4	A revised version of most of the paper
		7	A revised version of half of the paper
		5	A revised version of half of the paper
		6	Inspiration from the paper
	4	3	A revised version of most of the paper
		5	A revised version of half of the paper
		1	A revised version of most of the paper
III	5	Completely original	
	6	2	Some paragraphs
	7	Completely original	
	8	2	Some paragraphs
IV	9	All publications	Parts of the conclusions

*There are things known and there are things unknown, and in between are the doors of perception.*

—Aldous Huxley

# 2

## Research background

### Contents

---

<b>2.1</b>	<b>Socio-Technical Attacks . . . . .</b>	<b>11</b>
2.1.1	Socio-Technical Attack (STA) without <i>malicious intent</i> . . . . .	13
2.1.2	STA with extraneous <i>malicious intent</i> . . . . .	15
2.1.3	STA with user’s <i>malicious intent</i> . . . . .	17
<b>2.2</b>	<b>Socio-Technical Security Analysis . . . . .</b>	<b>18</b>
<b>2.3</b>	<b>Different audiences with different goals . . . . .</b>	<b>20</b>
<b>2.4</b>	<b>Inspiration from the safety field . . . . .</b>	<b>21</b>
<b>2.5</b>	<b>Research Questions and Objectives . . . . .</b>	<b>22</b>

---

*In this chapter, we present and define several concepts used throughout this thesis along with the research revolving around them. The first concept to be presented is that of Socio-Technical Attack, or the attacks that are possible because of the presence of human components in a system. Thereafter, we present research works relevant for socio-technical security analysis and thwarting Socio-Technical Attacks. Next, we present how the safety field tackled the problem of incorporating human aspects into the analysis of safety incidents, and how it could inspire us to improve upon socio-technical security analysis. Finally, we detail the research questions and objectives that this thesis tackles.*

### 2.1 Socio-Technical Attacks

As stated in introduction, assuring that a system is effectively secure is a complex matter because some attacks target not only the system’s technical aspects but also

the system’s users. We call these attacks Socio-Technical Attacks (STAs). In order to characterise the term with precision, we need to first discuss the following related concepts:

**Definition of Attack** An action that leads ‘to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of anything that has value to the organization’ (adapted from [73]).

**Definition of Threat** ‘Potential cause of an unwanted incident, which may result in harm to a system or organization’ [73].

**Definition of Vulnerability** ‘Weakness of anything that has value to the organization or [weakness] of means of managing risk<sup>1</sup> that can be exploited by a threat’ [73].

**Definition of Socio-Technical Attacks** ‘Attacks that exploit vulnerabilities that arise from human behaviour in conjunction with technology’ (quoted from David *et al.* [47]).

The research that gravitates around STAs is rich and tackles different aspects of the impact humans have on security through different strategies. For instance, defences against attacks which exploit users’ misunderstandings of a system will be radically different from defences against attacks where users willingly violate security policies. In one case, the research should focus on usability and training, and in the second case, the research should focus on policy compliance, user motivations, and business processes. For this reason, we propose to present the research related to STAs through the lens of a classification that we present below.

STA exploit the human to harm a system’s security. But humans are not necessarily unaware of harming the system’s security, and hence, can be honest or malicious. Therefore, we can classify STAs in several categories that differ in terms of user intent. If the behaviour that empowers the STA is the result of the user’s *malicious intent*, then the user is the attacker; however, if there is no malice in the user, and instead, the malice is in someone else using the user as a proxy without the user being aware of it, then we consider the user to be good intentioned. We resume our categorisation of STAs into the following three different types of STA:

**Human Error** STA, in which there is no *malicious intent*,

**Insider Threat** STA, where the *malicious intent* lies in the user,

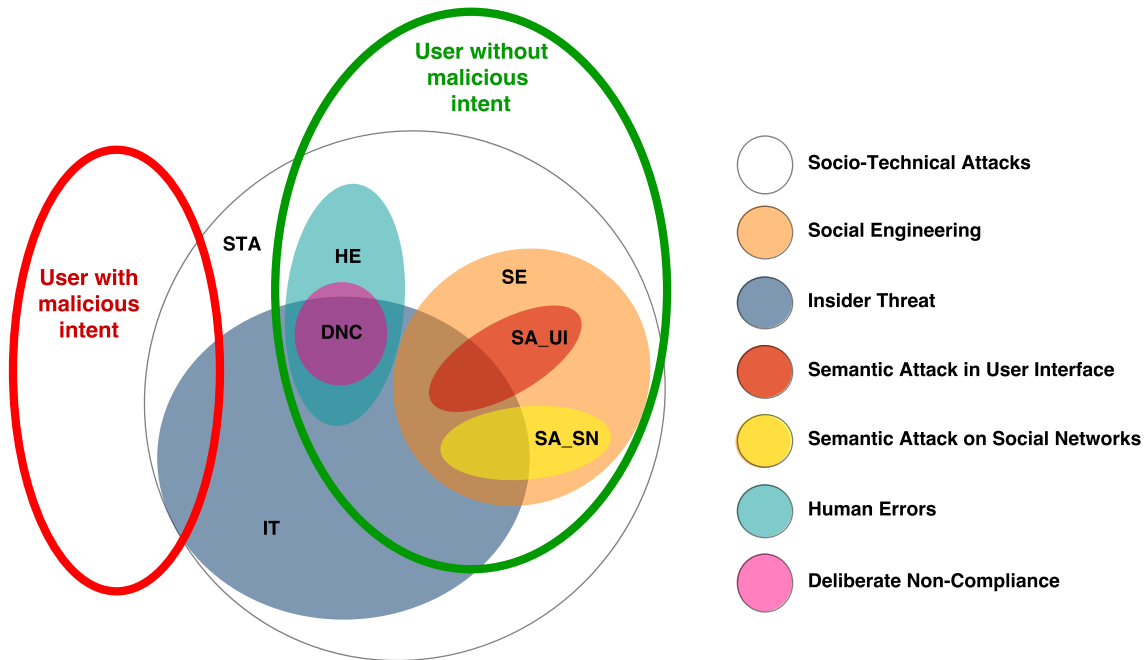
**Social Engineering** STA, where the *malicious intent* is extraneous to the user.

For each type of STA we give a *definition*, the *scope* of the attacker’s actions for performing the attack, and a non-exhaustive list of *examples* and *research* works about understanding and classifying these attacks. The different STAs are not mutually exclusive. In Figure 2.1, we have represented the different categories of attacks and their relation with the good/bad intention of the user. This picture shows that Insider Threat has overlaps with the other type of STAs, albeit we present Insider Threat only from the perspective of a malicious insider.

---

<sup>1</sup>‘Means of managing risk’ actually stands for ‘controls’.

Figure 2.1: This figure shows the overlaps between the different STAs presented. Semantic Attacks are part of Social Engineering. Deliberate Non-Compliance is part of Human Error. Insider Threat overlaps with Social Engineering attacks and Human Errors, and malicious users are only part of Insider Threat related research. The ‘Well Meaning Users’ area comprehends STAs without *malicious intent* and STAs with extraneous *malicious intent*.



## 2.1.1 STA without *malicious intent*

### 2.1.1.1 Human Error

Human Errors happen in every human activity and the use of computers is no exception. Lee [123] studied the publications of the (UK) Information Commissioner’s Office related to data breaches and found that 44.4% of data breaches mentioned in these publications are due to Human Errors.

**Definition** ‘an action or decision that results in one or more unintended negative outcomes’ (quoted from [163]).

**Scope** —non applicable—

#### Examples

- Configuration error: the user commits a configuration error. For instance, an Internet service provider’s network administrator who misconfigures the Border Gateway Protocol rules of an Internet router that subsequently causes a massive Internet outage.

- Bad recipient: the user sends a confidential message to a wrong recipient by entering a wrong address in the ‘to’ field of an email.
- Send wrong key: the user mistakenly sends a message along with is private key to a recipient, jeopardising the key’s confidentiality.

**Research** Research on Human Error outside of security is abundant, the most prominent work being Reason’s Generic Error Modelling System [147]. Within security, Miyamoto *et al.* [132] established a human error database for security. Furthermore, several schema intended for describing security incidents offer lists of possible human errors linked to a security incident (see for instance VERIS [176]).

### 2.1.1.2 Deliberate Non-Compliance

As users can harm a system’s security, organisations set security policies to delineate the frontier between acceptable and unacceptable use of the system. A security policy, is ‘a statement that defines the security objectives of an organization; it has to state what needs to be protected; it may also indicate how this is to be done [79]’. More specifically, security policies define objectives, or ‘statement of intent to protect identified resources from unauthorized use’ [162]. Security policies are organisational at a first level (‘a set a rules, laws , and practices that regulate how an organisation [...] achieves a security policy objective’), then at a second level, they are automated, or technically implemented as ‘a set of restrictions and properties that specify how a computing system prevents information and computing resources from being used to violate and organizational security policy’. Deliberate non-compliance occurs when a user chooses to break the rules or circumvent a security policy, with for instance, the intent to take a shortcut to a business goal.

**Definition** ‘Policy violations that: (1) knowingly break rules (employees violate security policies that they know exist); (2) are voluntary (actions are not forced by other parties e.g., supervisors); (3) are intentional (employees make conscious decisions to engage in the actions); and (4) are non-malicious (employees are not trying to cause damage)’ (quoted from [83]).

**Scope** Business processes and security policies.

#### Examples

- Healthcare: security policies are often circumvented in healthcare, and in particular, with regard to the use of passwords and other door access codes. Passwords and access codes tend to be shared between users by taping them onto medical devices, computers, and emergency supply rooms [115].
- Ecological consideration: a user can decide to put confidential documents in the recycle bin against a security policy that requires shredding because of the intention to protect the environment.



- Personal use of a device: a user can set up a weak password to unlock his own device for the sake of convenience, thus circumventing the organisation's security policy that rules how the organisation's assets should be accessed and protected from the user's own device (Bring Your Own Device) [92].

**Research** Research shows that deliberate non-compliance is often the result of ill-designed security policies that conflict with the user's goals [15], the organisation's business process [72], or both. By exploring the reasons for non-compliance, Kirlappos *et al.* show that security policies should reconcile the user's primary task with the business processes and security to provide effective security [112]. Later, Kirlappos *et al.* show that security practitioners can draw lessons from what they call 'Shadow security,' or the user's own alternative to an organisation's security policy, to build security policies that 'fit with the organisation's business, rather than impede it.' [113]

## 2.1.2 STA with extraneous *malicious intent*

### 2.1.2.1 Social Engineering

There is a plethora of literature that relates to Social Engineering (SE). The main reason is that SE is closely related to influence, deception, persuasion, and other umbrella notions that are connected with activities that are capable of influencing human decision-making. As Mouton *et al.* state in their work on the definition of the SE domain in security [134], there are a lot of different definitions for SE, however, they all can be summed as:

**Definition** 'The science of using social interaction as a means to persuade an individual or an organisation to comply with a specific request from an attacker where either the social interaction, the persuasion or the request involves a computer-related entity.' (quoted from [134])

**Scope** Social Engineering focuses on the Human.

#### Examples

- Baiting: the attacker plants a usb stick infected with a malware in the vicinity of an organisation (e.g., its parking lot) with the hope that a user plugs the device into one of the organisation's computers [161].
- Phishing: masquerading as another entity, the attacker contacts a user. This attack can be produced on numerous mediums (email, Internet Forum, SMS and the like) and usually aims at redirecting users to a malicious website for acquiring credentials [70].
- Pretexting: the attacker pretends to belong to a trusted group (e.g., technical support) and asks the user to give away confidential information or to perform actions that benefit the attacker. Pretexting attacks usually aim at justifying the attacker's physical presence in a restricted area to later obtain credentials or network access.

**Research** The most prominent work on SE is Mitnick’s seminal book [131]. Mitnick presents several examples of SE along a generic framework for SE Attacks. Later on, several researchers built taxonomies or schemas of SE attacks, among which we have consulted the works of: Harley [85], Janczewski *et al.* [103], Larabee [120], Ivaturi *et al.* [101], Tetri *et al.* [169], Algarni *et al.* [7], and Mouton *et al.* [134].

There is also a host of research focused on understanding how SE Attacks, frauds, and scam work, and what are the principles that make these successful. Several taxonomies of principles are often referred to in the literature: Cialdini’s 6 principles of persuasion [34], Stajano *et al.*’s 7 psychological principles of scams [159], and Gragg’s psychological triggers [81]. Ferreira *et al.* studied the overlap between these different principles and their use in phishing [62].

### 2.1.2.2 Semantic Attacks in the User Interface

Semantic Attacks are a special kind of SE Attacks that focus on the User Interface. The term was first proposed by Schneier in 2000 [153]. Later, Heartfield *et al.* [86] proposed the following definition:

**Definition** ‘The manipulation of user-computer interfacing with the purpose to breach a computer system’s information security through user deception.’

**Scope** Focuses on the User Interface.

#### Examples

- Web Pop-up: the attacker produces a pop-up while the user is navigating on a website, thus abusing the trust that the user has on the website. The attack usually collects information, for instance login credentials and credit card information.
- Rogue Wi-Fi Access Points: the attacker creates a Wi-Fi access point with the same name (SSID) as another one with the hope to bait some user to connect to it and launch a man in the middle attack.
- Phishing: introduced earlier as an example of SE attack, phishing is also a perfect example of Semantic Attack.

**Research** Several works attempted to classify Semantic Attacks. Mohd *et al.* [133] built a taxonomy of SE attacks where semantic attacks are listed under the category ‘technical-based social engineering attacks’. Heartfield *et al.* [86] built a taxonomy of Semantic SE Attacks and a Survey of defence mechanisms. Sood *et al.* showed how Semantic Attacks can be used to deliver malwares [156].

### 2.1.2.3 Semantic Attacks in Social Networks

Another definition of Semantic Attacks is tied with the spread of *misinformation* and *disinformation* through social networks. Kumar *et al.* [116] propose the following definition:

**Definition** ‘Semantic attacks are aimed at influencing the perceptions of users with the aim to modify their actions.’

**Scope** Focuses on the information perceived by the user.

### Examples

- Sybil Attacks: an attacker creates several identities on a social network to spread false information, with the goal of influencing that platform’s users.
- Shill Attacks: several users of a social network agree to push an information forward (or rate it higher) to give it more weightage, and in turn, abuse the social network’s recommender system into relaying this information to other users. Again, the attacker’s goal is to influence the chosen social network’s users.

**Research** Most research related to this kind of Semantic Attacks (and their classification) are linked to recommendation systems or to the propagation of rumors, and hence, fall out of scope of this thesis.

## 2.1.3 STA with user’s *malicious intent*

### 2.1.3.1 Insider Threat

Insider Threat is<sup>2</sup> a type of STA in which a system’s user is not exploited by an attacker, but the user himself is the attacker. The user willingly chooses to misbehave for one’s benefit (e.g., the user himself). Contrary to an external attacker, an insider is already within the perimeter and possesses knowledge as well as credentials. His malicious activities can hardly be distinguished from his harmless activities.

**Definition** ‘An insider threat is [posed by] an individual with privileges who misuses them or whose access results in misuse.’ (quoted from Hunker *et al.* [94])

**Scope** Every aspect of an organisation can play a role in triggering this STA, such as policies (e.g., security policies), organisational culture, management, user’s predisposition, and the like.

### Examples

- Disgruntled employee: a former employee who has recently been fired may use his credentials that have not been revoked yet to take revenge.
- Whistle-blower: an employee, who has ethical issues with the organisation’s policies and activities, uses the credentials that his manager shared with him to ease everyday work, to exfiltrate confidential data and then leaks it to the press.

---

<sup>2</sup>Again, we reduce Insider Threat to STAs where the user has a malicious intent for the sake of their introduction. But, as mentioned previously and depicted in Figure 2.1, Insider Threat is not limited solely to this sole type of attacks.

**Research** Band *et al.* [13] studied the psychological, technical, organisational, and contextual factors that turn an insider into an insider threat. Alfawaz *et al.* [6] developed a framework to classify Insider Threats from the perspective of insider's knowledge, skills, and individual preferences. Magklaras *et al.* proposed a tool to evaluate the probability of IT misuse from certain profiles of user behaviour [126]. Gavai *et al.* [76] identify abnormal behaviours after analysing online activities.

## 2.2 Socio-Technical Security Analysis

After having presented several types of STAs along with bodies of research that aim at understanding and categorising them, we now move forwards and discuss works that intend to defeat STAs. To thwart STAs one needs to identify which factors make a human-computer interaction turn into a socio-technical vulnerability. This leads us to our definition of Socio-Technical Security Analysis:

**Definition of Socio-Technical Security Analysis** ‘A security analysis that aims at identifying vulnerabilities that arise from the human behaviour in conjunction with technology.’

The study of socio-technical systems and their security is inherently transdisciplinary [185]. Indeed, scientists with different backgrounds need to contribute or cooperate to study the security of socio-technical systems. Social sciences, cognitive sciences, computer sciences, and human-computer interactions are, for instance, very relevant when it comes to the study of the whole spectrum of interaction between a system and its users, which can lead to the revelation of socio-technical vulnerabilities [16].

Coming from different backgrounds, scientists naturally use different approaches when tackling the problem of analysing the security of socio-technical systems. The convergence of these different fields of research towards a common subject of analysis is taking place from different perspectives one paper at a time. Computer scientists tend to take a systemic approach with a tinge of social sciences, while other scientists tend to apply the methods of their field to tackle immediate issues.

In the following section, we give an overview of the works that focus on socio-technical security from the perspective of computer sciences and social sciences, without attempting to categorising them. Eventually, we introduce the works that propose the concepts that, we think, are the most integrated approaches from different disciplines in their take on socio-technical security analysis: security ceremonies, and the *concertina model*.

Kainda *et al.* [110] evaluate a list of usability variables (e.g., satisfaction, facility of use, errors committed) of different protocols used to establish out-of-bound channels between human peers, and how they relate to the robustness of the protocols against a Dolev-Yao adversary [52].

There are few works in computer sciences that aim at using formal methods to verify human-computer interactions. For instance, Curzon, Rukšėnas and Blandford's work on this matter, first generally [45], then applied to security [149].

Dalpiaz *et al.* [46] develop a Socio-Technical Security modelling language that specifies the security and trustworthiness requirements for cross-organisational systems. Further, they develop a tool to support the language [140, 139].

Some works also analyse the security of Socio-Technical Physical Systems that analyse the security of physical systems while incorporating human agents. For instance, Dimkov *et al.* [51] create ‘Portunes’ to describe attack scenarios that span across the physical, digital, and social domain.

Works about browser warnings conclude that users do not look at browsers’ cues or security indicators, and that they are led to incorrect decisions 40% of the time [50]. Sunshine *et al.* [165] further conclude that even custom/improved warnings are not enough, and ideally, security designers should avoid them altogether. Research on warnings is still ongoing and is very active. The most prominent work appears to be Felt *et al.*’s work on Google Chrome’s warnings [4, 61].

Usability is also a field of research that is investigated when it comes to the analysis of socio-technical security [10]. Kainda *et al.* [111] study how methods used in Human Computer Interaction (HCI) to improve users’ effectiveness, efficiency, or satisfaction can be used while still preventing the introduction of potential new vulnerabilities through their use. Further, they create a threat model for usable security. One prominent subject of research for usability of security systems is Pretty Good Privacy (PGP): Whitten first studied the (un)-usability of its most used client in 1999 [184], Ruoti *et al.* update this research by studying modern PGP clients [150], concluding that ‘PGP is still unusable for the masses.’

The way users accept security mechanisms and policies is also a subject of research. For instance, Zhang *et al.* investigate the impact of passwords expiration and replacement [192] and confirm what Adams and Sasse identified in 1999 [3]: expiring passwords is a questionable practice because users tend to generate new passwords from the old ones, weakening the system’s security. Another study was conducted in laboratory experiments on password generation strategies [173], which observed that people are not good at generating unpredictable passwords, and that additional research should be conducted to fix this problem.

Psychology is studied, in particular, to study the user susceptibility to attacks. For instance Downs *et al.* [54, 53], Sheng *et al.* [155], and Bowen *et al.* [23] study users’ susceptibility to phishing attacks.

Cranor *et al.* [12, 42] propose a framework to understand how security failures happen when users misbehave because of flawed human-computer communications. This framework is a sequence of generic steps a designer follows to identify potential failure points for each technical function of the system in which the user participates. The designer needs to mitigate those failures, either by eliminating user’s intervention altogether, if possible, or by improving user interaction.

Conti *et al.* [40] research on visualisation systems that typically include the human in the decision-making loop and present a visual taxonomy to identify attacks. Falk *et al.* [59] examine the prevalence of user-visible security design flaws in financial websites with high security requirements, and present a methodology for testing these issues by selecting the five most common security user-visible flaws of website design and identifying them in a set of websites.

Camp *et al.* [28] study how mental models associated with the perception of risk could help the user to be more robust against STAs. Volkamer *et al.* review how mental models could be used to ‘design security solutions and interactions more effectively.’ [178]

In the last 15 years, several works have contributed to extending the scope of the analysis of security protocols in order to incorporate what was until now considered as out-of-band: the human component. It started with Blaze in 2004 with the notion of ‘Human-scale security protocol’ [20], and Ellison in 2007 with the idea of ceremonies in which not only technical agents but humans could also be a part of the analysis [56].

Then Bella and Coles-Kemp [16] created the *ceremony concertina*, which models human-technology interactions as a set of layers encompassing, for instance, the network, the operating system’s processes, the User Interface, the user’s persona, and ultimately, the society. Each layer can be folded or unfolded to focus on some part of the interaction. As Bella *et al.* state in [17]: ‘The ceremony concertina model offers anyone from a range of disciplines, who wants to investigate socio-technical aspects of security and privacy, a canvas on which to paint their findings.’ In other words, the *ceremony concertina* offers a model through which scientists can express their problems, hypotheses, and use their own methods.

To sum up, security ceremonies expand the study of security protocols to include the humans and *ceremony concertina* provides a reference model to study all the elements between users and technology. This allows for the study of the human aspects of a security protocol with social sciences methods concurrently or subsequently to a traditional formal analysis of this protocol, within the same reference model. This property of security ceremonies and *ceremony concertina* hint that they could be a useful starting point for a framework to study the social and technical aspects of socio-technical security analysis.

## 2.3 Different audiences with different goals

We identified two communities of professionals that are interested into thwarting STAs or reusing the knowledge gained from the previously presented research works: security researchers, and security practitioners. Alas, they may have different goals, constraints, and skill sets.

For instance, security researchers abide by scientific rigour and use scientific methods to produce results that are meant to answer specific research questions as truly and extensively as possible. Security researchers can choose an issue, investigate their hypothesis regarding the issue, and go to great lengths to ensure the significance of their results.

Security practitioners, on the other end, intend to solve a very different problem. They can, for instance, be in charge of securing the operations of a whole company, or investigate a security incident that occurred in a complex environment. Furthermore, security practitioners establish security on a budget and are constantly trading off security for cost optimisation. They may not be interested in answering a research question extensively, rather they are likely to be satisfied to produce results that are ‘good enough’ to effectively secure businesses’ operations.

These differences in goals and constraints between security researchers and security practitioners hint that different methods should be produced to fulfil their respective needs. Whereas security practitioners can be more interested in a tool that can effectively identify potential STA, security researchers need a framework in which they can produce reliable results to advance their understanding of STAs.

## 2.4 Inspiration from the safety field

There are the following two main research fields in safety: Human Factors (HF), which is a synonym for ergonomics<sup>3</sup>, and Human Reliability Analysis (HRA) which is the predictive and retrospective study of human errors. HRA gives ground to Probability Risk Assessment (PRA) by identifying what can go wrong and how an event can unfold depending on human performance, thereafter, PRA quantifies the likelihood of failure and of the potential aftermath.

Safety is a field of research from which security borrows a host of methods and concepts. For instance, most of the methods used for testing usability (e.g., cognitive walkthrough, focus groups, think aloud protocol, *et cetera*) are borrowed from the HF field of research<sup>4</sup>. This is because the safety field realised more the 60 years ago that a system could not be safe without encompassing the human component in its analysis, and thereby, created methods and techniques to do so [32].

However, the resemblance does not stop here. Indeed, we can draw a parallel between safety HF's field of research and works that take a social sciences approach to security analysis, as both these research works draw on psychology and empirical methods to improve human performance. In the same manner, we can relate HRA, which draws on system design and engineering, to the works that take a computer sciences approach to security analysis. For instance, PRA techniques use event-trees and fault-trees, which inspired the security field into creating attack-trees [128].

Another parallel is the struggle both the fields have to go through to utilise social sciences' methods and findings when analysing the security or the safety of systems. In a retrospective paper [21], Boring contemplates the different trajectories HF and HRA have taken in the course of the last 50 years, and how they are interconnected. While both fields stand upon psychology, HRA is aligned with reliability engineering and uses generalised empirical data (i.e., error modes, performance shaping factors, *et cetera*) to operate, whereas, HF embraces empirical studies and is used primarily in system design. Thus, HF tends to be more up-to-date in terms of findings on human performance, and it constitutes readily available insights on human performance for HRA. We argue that the relation between the social and technical studies of security is the same, albeit the generalised form of social sciences findings regarding socio-technical security is yet to be defined.

Indeed, most of the research works that we presented in Section 2.2 rely on literature reviews to identify which aspects pertaining to the user should be considered before undertaking their technical work (for instance, Carlos *et al.* with [30] and [29]). The field needs a methodology that creates a pipeline, which align social

---

<sup>3</sup>Also called HFE for HF and Ergonomics.

<sup>4</sup>Which, to be fair, borrowed a lot from psychology.

sciences findings with consequences on a system's security, just as HRA is doing for systems' safety in reliability engineering. Furthermore, we reckon, the field will face the same struggle that HRA and HF have been facing for 50 years. The analysis of socio-technical systems will need a way to keep insights from social sciences up to date.

## 2.5 Research Questions and Objectives

In Section 2.1, we saw that STAs are diverse, and that this diversity is reflected in the literature related to the classification and explanations of these attacks. Some STAs, such as phishing, are well-known and have been investigated thoroughly, while others, such as the exploitation of human error, are acknowledged but remain mainly out-of-scope in security research.

Thereafter, we highlighted different research works that explore the same problem, i.e., how to deter STAs. These works question different aspects of these attacks. For instance, some research works focus on how the communication between the system and its users fails to convey effectively a message critical for the system's security (i.e., [12, 42, 4, 61, 50, 165]) while others focus on psychological aspects that could undermine security (i.e., [54, 53, 155, 23]). We propose to tackle the problem of deterring STAs and of identifying socio-technical vulnerabilities through the study of three research questions.

### Research Questions

- RQ<sub>1</sub> How can we detect a socio-technical vulnerability in a system?*
- RQ<sub>2</sub> How can we identify in the interactions between a system and its users, the human behaviours that can harm this system's security?*
- RQ<sub>3</sub> How can we identify the factors that foster human behaviours that are harmful to a system's security?*

The first research question, *RQ<sub>1</sub>*, is the main crux of the thesis. It is the outcome of a successful socio-technical security analysis. *RQ<sub>1</sub>* aims at providing a unified way to identify the factors that could foster the human behaviours that are harmful for a system's security, and to identify which factors should be considered when adding a human element to a system. The study of this first research question leads to the study of the second and the third. To detect socio-technical vulnerabilities, one first needs to identify the human behaviours that have adverse consequences for the system's security (*RQ<sub>2</sub>*), and then determine which are the factors that foster these behaviours (*RQ<sub>3</sub>*).

We stated in Section 2.3 that we can identify two different communities that have reasons to be interested in studying and thwarting STAs: security researchers and security practitioners. For each community, we identify an objective that this thesis aims at fulfilling as its contributions:

### Objectives



- $O_1$  *What form should a framework that intends to identify harmful behaviours for security, and to investigate the factors that foster their occurrence take?*
- $O_2$  *What form should a semi-automatic, or tool-assisted methodology for the security analysis of socio-technical systems take?*

The objective  $O_1$  aims at supporting security researchers in the study of these research questions. As explained in this chapter, the study of these research questions poses methodological challenges. Indeed, being able to predict the consequences of the addition of a human component to a system requires knowledge and methods from both computers sciences and social sciences. Computer sciences are needed to ponder upon the consequences of a behaviour, and social sciences to identify what factors in this system could trigger this behaviour. Bella and Coles-Kemp's *concertina model* [16] constitutes the canvas on which we draw our answer to  $O_1$ . We create a framework that works as a common ground for socio-technical security analysis. This framework supports the resolution of  $RQ_2$  and  $RQ_3$  by providing ways of identifying security-critical behaviours as well as ways of investigating the factors that foster their occurrences.

The objective  $O_2$  aims at supporting security practitioners and providing an answer to the first research question  $RQ_1$ . Inspired from HRA methods in safety, the methodology we propose to answer  $O_2$  is a Root Cause Analysis (RCA) technique tailored for socio-technical security analysis. This method will support the resolution of  $RQ_1$  by generalizing results from social sciences studies and reusing these results into socio-technical security analyses.



## Part II

# STEAL, a framework for socio-technical security analysis, and applications



*Toutes les voies sont barrées, il faut pourtant agir. Alors nous essayons de changer le monde, c'est-à-dire de le vivre comme si les rapports des choses à leurs potentialités n'étaient pas réglés par des processus déterministes, mais par la magie.*

—Jean-Paul Sartre, 'Esquisse d'une théorie de l'émotion'

# 3

## STEAL: a framework for socio-technical security analysis

### Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>28</b>
<b>3.2</b>	<b>The STEAL framework</b>	<b>29</b>
3.2.1	A socio-technical security conceptual framework	29
3.2.2	Illustrating the use of STEAL: Wi-Fi networks selection	34
3.2.3	Summary	37
<b>3.3</b>	<b>STEAL validation: social focus</b>	<b>37</b>
3.3.1	Introducing the study	38
3.3.2	Use Case Scenario	39
3.3.3	The Survey	40
3.3.4	Context	41
3.3.5	Security Discussion	45
3.3.6	Conclusion	45
<b>3.4</b>	<b>STEAL validation: technical focus</b>	<b>46</b>
3.4.1	Introduction	47
3.4.2	Study	48
3.4.3	Related Work	50
3.4.4	Conclusion	50
<b>3.5</b>	<b>Discussion</b>	<b>51</b>
3.5.1	Reference Model	51

3.5.2 Support to social and computer sciences methods for security analysis . . . . .	51
<b>3.6 Conclusion . . . . .</b>	<b>52</b>

---

*In this chapter, we introduce STEAL, a framework to study socio-technical security. The main goal of STEAL is to provide computer scientists and social scientists with a common ground and a similar terminology to allow for a security analysis that comprises technical, social and contextual elements. In the first part of this chapter, we present the framework, its reference model and associated methodologies. Then, we move on validating its relevance as common ground for socio-technical security analysis through two studies: the validation of TLS certificates in modern web browsers, and the selection of Wi-Fi networks. Part of the content of this chapter appears in three papers: ‘A Conceptual Framework to Study Socio-Technical Security’ by Ana Ferreira, Jean-Louis Huynen, Vincent Koenig and Gabriele Lenzini published in the Proceedings of the second International Conference on Human Aspects of Security 2014, (HAS 2014, Heraklion, Crete, Greece) [65]; ‘Studies in Socio-technical Security Analysis: Authentication of Identities with TLS Certificates’ by Ana Ferreira, Rosario Giustolisi, Jean-Louis Huynen, Vincent Koenig and Gabriele Lenzini published in the Proceedings of the 12th International Conference on Trust, Security and Privacy in Computing and Communications (Trust-Com 2013, Melbourne, Australia) [63]; ‘Socio-Technical Study on the Effect of Trust and Context When Choosing WiFi Names’ by Ana Ferreira, Jean-Louis Huynen, Vincent Koenig, Gabriele Lenzini and Salvador Rivas published in the Proceedings of the 9th International Workshop on Security and Trust Management (STM 2013, Egham, UK) [68]; and ‘On Tools for Socio-Technical Security Analysis’ by Ana Ferreira, Rosario Giustolisi, Jean-Louis Huynen, and Gabriele Lenzini presented at the Grande Region Security and Reliability Day (GSRD 2013, Luxembourg) [64]. The research presented in Section 3.4 was originally completed by Rosario Giustolisi and reported on in two papers I have co-authored with him where these results are linked to the STEAL framework (see [64] and [63]). The aforementioned work also appears in Dr. Giustolisi’s PhD thesis.*

## 3.1 Introduction

Systems that are secure even when used by humans —a property that we call *effective security*— are hard to validate and evaluate. A system can embed technical mechanisms that make it technically secure, such as encryption protocols, but those mechanisms can fail if users bypass or misuse them. Such failures are common since humans do not perceive security as a primary goal [183] [187] and do not properly assess risks when using information communication technology [171, 117].

There is more: as explained in the previous chapter computer system designers, with a few exceptions [142] [111] [42] [29], are not accustomed to count human cognitive and behavioural traits as risk factors in the security requirements. Thus, even systems that have been validated as technically secure, may still be insecure against STAs (e.g., social engineering).

How can we achieve a better effective security? There is no once-and-for-all solution. Effective security is a complex quality to achieve. It is inherently socio-technical (it depends on how human and technical aspects integrate) and it may be context and culture (including education) dependent [168, 179]. For example, in hospitals, access control solutions cannot be effective unless designed to fit the nomadic, interrupted, and cooperative nature of the medical work [14]. But, the same access control solutions would be judged differently in a context such as a bank, where employees work mostly alone and where security requirements must consider, for example, threats coming from hackers (e.g., see [182]).

To make a system effectively secure in different scenarios, it likely requires diverse strategies and solutions. However, it is possible to refer to a common framework of analysis. Such a framework should help computer security designers and social scientists to collaborate by providing an operational guideline for an interdisciplinary approach in studying a system's security, as well as tools and methodologies for questioning security at both the technical and the social layers.

**Outline.** This chapter proposes and describes a common framework for socio-technical security analysis that we call STEAL (*Socio-TEchnical Attack AnaLysis*). First we introduce STEAL, its reference model and the methodologies that it supports. Then, we assess STEAL's relevance by investigating: if (1) STEAL's reference model is adequate for supporting both social and technical analyses of systems, and if (2) STEAL's methodologies and operational guidelines can be instantiated and are relevant in studying socio-technical security. We investigate these research questions by studying two use cases: the selection of Wi-Fi networks and the validation of TLS certificates in web browsers. The first use case (Wi-Fi) focuses on validating the practicality of performing social analyses of security and the relevance of STEAL's reference model. The second use case (TLS) focuses on validating the use of technical analyses for socio-technical security. Eventually we discuss STEAL achievements and limitations.

## 3.2 The STEAL framework

### 3.2.1 A socio-technical security conceptual framework

By a socio-technical security conceptual framework, we mean an operational guideline for a systematic approach in modelling and analysing a system's security in its technical and social perspectives. Past research in security validation shows that important elements of such a framework are (I) a *reference model* and (II) a set of procedural *methodologies*. (I) is to describe, at a suitable level of abstraction, the elements of the system that we intend to analyse. (II) is to have tools for a technical and a social experimental analysis of security.

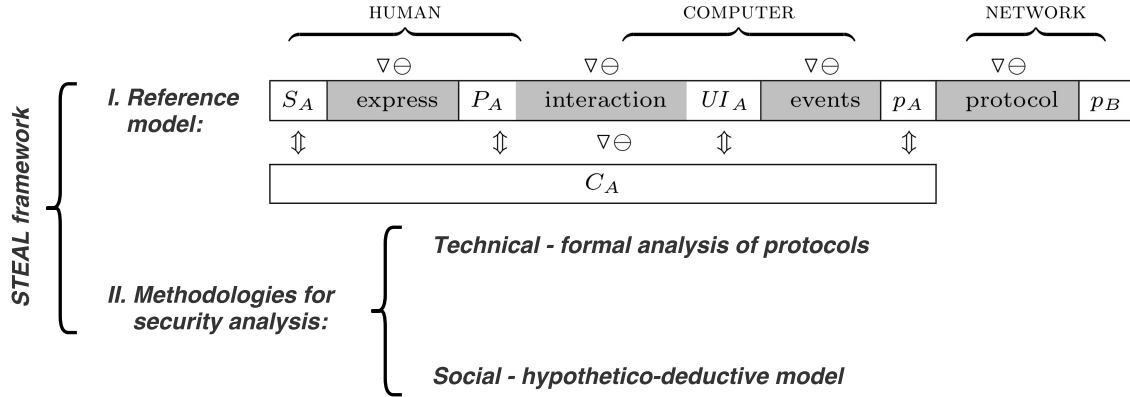


Figure 3.1: STEAL Conceptual Framework.  $\nabla$  are possible attacks, and  $\ominus$  are possible defences. The attacker can strike in the context, and in every layer between the user and the network.

STEAL, our framework, includes them both (see Figure 3.1). Its reference model (see Section 3.2.1.1) suggests a system composed by interacting elements/actors (human, interfaces, processes, and context). Its set of methodologies (see Section 3.2.1.2) includes security validation procedures coming from the formal analysis of security protocols and from the applied cognitive sciences and usability research.

### 3.2.1.1 STEAL: reference model

The reference model is a variant of the Bella *et al.*'s [16] concertina model (Figure 3.1, upper part). A socio-technical system is abstractly seen as layered, each layer made of communicating/interacting elements. There is at least a human persona, say Alice ( $P_A$ ), and the technology she is using. This is further composed by at least a human interface ( $UI_A$ ) and some software processes ( $p_A$ ). Processes can, through a network, communicate with other processes ( $p_B$ ), behind which may stay one or more humans, say Bob ( $P_B$ ), who are in turn interfaced to human interfaces ( $UI_B$ ). Layers can be folded, with the effect that not all elements need to be necessarily in place. Representing our system in this way helps the analyst to select the key components for analysis, and to distinguish between the technical, the human components and the context.

STEAL extends this model by adding the context ( $C_A$ ), and attack and defence models. Context is the physical or social environment where the interactions for ‘Alice’ take place.  $C_A$  influences how  $A$ ’s *self* ( $S_A$ , in Figure 3.1) expresses into  $P_A$ ’s, the way  $P_A$  interacts with the interface, and the software, which can be context-dependent.  $C_B$  does the same on  $B$ ’s side, not shown in the figure.

This simple reference model fits many scenarios. For example, in an ATM machine scenario, Alice ( $P_A$ ) is the client, the user interface ( $UI_A$ ) is the ATM’s set, and  $p_A$  is the software executing the client instruction that connects the ATM with the bank ( $p_B$ ). The context ( $C_A$ ) is where the ATM is located, a street or the interior of a bank’s hall. In a scenario where Alice is accessing a protected web page, the web interface is ( $UI_A$ ), the browser is the process  $p_A$  that runs a protocol with the web server hosting the page, which is process ( $p_B$ ). The context  $C_A$  can



be Alice in her office, or in an airport’s hall. In a scenario with a few persons collaboratively editing a file in the cloud, the persons are the Alices and Bobs, their screens and keyboards the human-computer interfaces; the software they use to edit and to browse are the processes. The communication happens via the cloud service. The context can be where those persons are, at work, at home, the latter being not only the location but also social environments.

*Attack and defence models.* STEAL comes also with an attack and with a defence model. They are both relevant for the security analysis, as security is always evaluated with respect to an attacker with specific capabilities (respectively, a defender with specific capabilities). The icons  $\nabla$  (attacks) and  $\ominus$  (defences) indicate where the model assume attacks can strike and where defences can act.

Whatever the nature of the channels and the messages they carry, an attacker can intercept, modify and inject messages in any of those channels. These are typical abilities ascribed to a Dolev-Yao intruder [52]. However, differently from classical Dolev-Yao, in STEAL, the attacker controls not only the network but also the interactions between the application, the user’s interfaces, the persona, and the context. Therefore, an attack may be technical and or a social engineering kind of attack.

Defences also act by interfering with the communication channels. This includes the channels with the user. In our framework, users can participate to improve security, a substantial difference between our and other works [42].

*Other assumptions.* Our reference model assume that the observable behaviour of the system’s elements under analysis is (at least at the level of abstraction chosen) known. However, it does not assume, and does not depend on, the reasons, or the logic, behind this behaving be necessarily understood. This assumption endorses a computational approach. A component (whatever it is, human, interface, agent or context) is an entity (an automaton) that behaves according to a certain control logic that determines its input, output and internal *actions* depending on its *state* and on its (previous) inputs.

For example, a user at an ATM machine, behaves according to some beliefs, desires and intentions that he/she has (withdraw money) which, according to his/her state of mind (I have inserted a pin and wait for the money to come out), determine the actions he/she does (taking the money once out). In its turn the ATM machine’s logic is its software code, its state is the machine’s state (pin inserted, now checking it), and its actions (display selection of banknotes).

In practice, we may not be able to define precisely a component’s control logic, or to list the full set of actions it can ever perform, or to know the component’s state in time. But, to build a sufficiently consistent picture (i.e., model) of the component’s observable behaviour, one can apply indirect methods to inquire properties about an element’s state and to test propositions about it, or by observing the actions it does. For example, we can build a model of a browser by looking at its code. In this case we know fully how it works. If the code is proprietary, we may not be able to fully know its logic but we can build a consistent model by walking through its behaviour. Similarly, we can observe a user interacting with our browser, but we may not be able to observe him changing his mental state (e.g., cognitive process),

nor knowing why users behave in certain ways. We can only observe and ask him (e.g., questionnaire/interview).

This assumption is also motivated by the tools of analysis we are going to have: tools for a formal analysis such as model checkers, for the technical security, and human computer interactions methodologies, as those used in usability laboratories, for the social security.

### 3.2.1.2 Methodologies for socio-technical security analysis

STEAL has two methodologies for security analysis. One allows to understand the security properties without considering a complex model of user behaviour. The other allows to question hypotheses on human behaviour and on security properties with the human in the loop.

As shown in Figure 3.2, the two methodologies, together, make the socio-technical analysis possible. The technical analysis helps, against specific threats, discovering if attacks are possible. However, their effectiveness may depend on some user's decisions, exactly as it happens with TLS authentication (see Section 3.4), where a user may decide to proceed despite a warning flashing that the certificate is invalid. The experimental analysis answers whether those attack would be successful with real users and factual behavioural patterns. The outcomes of the social-oriented analysis also enlighten us on what factors influence critical decisions that may lead to attacks. Such outcomes may therefore suggest defences which, in turn, can be implemented at a technical or a social level or as a combination of them, and understanding their effectiveness triggers another round of analysis. Moreover, it is also possible to perform a security analysis against attacks purely against the human, like social engineering. At the current status of research there is not a stable theory able to model such attacks in a formal model way, thus to study their effect is again done experimentally. This can change in the near future.

To test hypothesis of user's behaviour under social attacks, we may need to launch such attacks and harvest data for analysis. This requires an authorisation from an ethical committee and a compliance with a legal framework, assurances that must strictly comply with ethical requirements. In certain situations this may be hard to achieve.

**Technical focus.** This methodology helps discovering whether an attack is present, within the defined threat model, and mostly with technical interactions and a simple user model. The technical security analysis is applied to elements from  $UI_A$  till  $p_A$  and possibly  $p_B$  till  $UI_B$ , including the context(s).  $P_A$  is modelled as a non-deterministic process i.e., interacting with process  $UI_A$  in every possible way [19, 18, 63]. The technical analysis, can use formal tools of protocol analysis (e.g., model checking [38]), with the only difference that communications are now multi-layered. In a simple case, the analysis can be pursued informally.

Analysing security in this focus means to verify whether specific security properties remain valid despite an intruder. The technical analysis may reveal vulnerabilities due to a faulty integration between the technical and the human layers, like it happens when a system does not offer users to change a password, when it

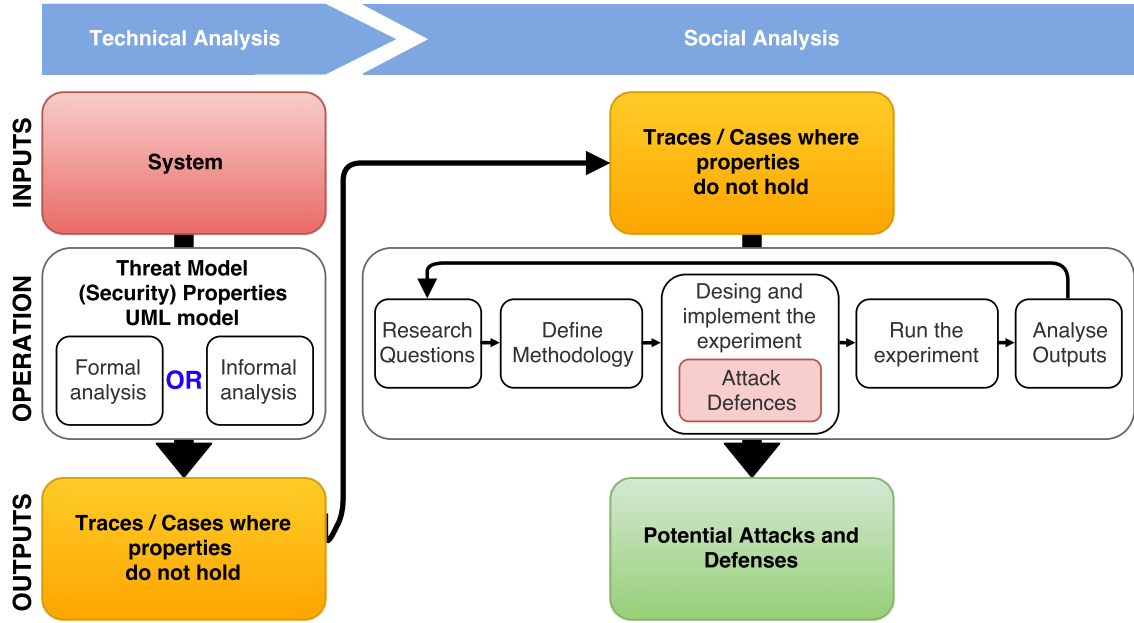


Figure 3.2: Operational guideline to use Technical and Social Analyses in a pipeline.

should (e.g., [16]). The output of the technical analysis gives ground for a successive security analysis with social-focus, as it provides information about what attacks should be considered there.

**Social focus.** This methodology helps discovering security failures in the human interactions, when a predefined threat model is present, or in presence of specific attacks revealed by the technical analysis. The social analysis focuses on human behaviour and choices, therefore from elements  $S_A$  till  $UI_A$  and possibly their human-to-human interaction with  $S_B$  via  $UI_B$ , including the context(s). The social analysis uses the hypothetico-deductive model from empirical social sciences research [78] and Human Computer Interactions [121] (see *Social Analysis*'s operation block in Figure 3.2).

Briefly, the process starts with the initial definition of *research questions* to be tested. These usually come from previous literature review, insights either observed or hinted by human computer interactions, or from the technical security analysis itself. The process continues with the definition of the most appropriate *research methodology/ies* (i.e., laboratory experiments, interviews, surveys) to answer the research question. Here we also decide on the appropriate threat model and the layers that can be impacted in the reference model. This process is similar if we are testing defences. The next step is to *design and implement the selected methodology(ies)* with the goal of making this process reproducible over a series of experimental tests. After all is set and ready to start, the *experiment is run* and *output data is collected and further analysed*. Usually, data can be analysed using both quantitative (statistical tools can be used to analyse data and test previous defined research questions, and show how significant these are) and qualitative methods (qualitative data gath-

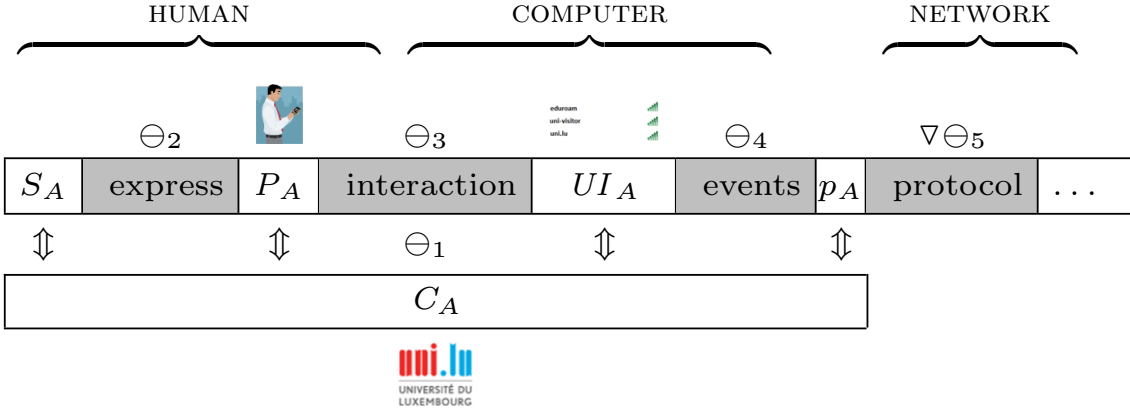


Figure 3.3: Reference model for WiFi connection to the Internet.

ered from the participants can be correlated with results obtained from statistical analysis and also provide insight or explanation on user’s behaviour).

In the next section, we illustrate the use of STEAL through the study of the selection of Wi-Fi networks. The example shows how this scenario can be described in the STEAL’s reference model, and how the socio-technical security analysis can be done by performing technical and social analysis in a pipeline.

### 3.2.2 Illustrating the use of STEAL: Wi-Fi networks selection

We describe how STEAL works with a scenario of a visitor at the Univ. of Luxembourg trying to get WiFi Internet access by choosing an *open* SSID name from the list he is presented by his device’s network manager.

#### 3.2.2.1 Reference model

STEAL reference model highlights the elements of the scenario (Figure 3.3), comprising the network manager and all the network communication protocols ( $p_A$ ), the interface on the user’s mobile device ( $UI_A$ ) and the user trying to select a wireless network name to connect to the Internet ( $P_A$ ). The premises of the University of Luxembourg, the place where all is happening, is the context  $C_A$ .

About the interactions, *express* would be the expression of all the human traits of a persona into how  $P_A$  takes security decisions when interacting with a human-computer interface in that particular scenario. (We are not able to model those expressions, but we may want to consider them in the analysis). Then, *interaction* are the actions performed by the user, to access the wireless network manager’s list and select an SSID name to connect; *events* are the communications exchanged between the user’s interface on the mobile device and the wireless network manager application and the wireless access point, which manages calls to its network; *protocol* are the network protocols and messages exchanged between the wireless network manager application and wireless access point, which manages all accesses to the services that its network provides.

### 3.2.2.2 Socio-technical security analysis

**Technical analysis.** We model the technical layers in a UML diagram. It illustrates the sequence of actions between those elements during an attack in this scenario (Figure 3.4). In theory it is possible to run a formal analysis against a Dolev-Yao attacker. Here, it is immediately evident that an intruder can open a rogue wireless access point because the SSID is not authenticated.

The success of the attack relies only on the user’s choice, precisely on whether a user will actually choose the rogue access point or not. This cannot be understood with this technical analysis only. However, we elaborate more on the attack before passing to the social analysis. We hypothesise that the context plays a very important role in this scenario as the attacker can use the University’s visual identity –and all that is connected with it such as knowledge, reputation, etc– to lure a victim to choose a rogue but meaningful name, such as “uni.lu”, over the University’s official SSID names (actually “uni-visitor” and “eduroam”). The attacker can also set up a second SSID, “secure\_AP”, a name recalling “security” and test which name has more appeal for the user. Figure 3.4 shows the attack.

**Social analysis.** To apply the hypothetico-deductive model for this analysis we devised the following stages: (1) *Research question*: do context and trust influence users’ choice of a wireless network name? do names reminding security influence that choice? (2) *Methodology*: online survey with two different groups of questions (one relating to context and the other to trust) each together with open questions to provide further explanation of the participant’s selection. The groups of questions must be answered by two different groups of participants (in a between subjects design) regarding wireless network names preferences and graded using a Likert scale (1 - less trusted/less preferred to 5 - highly trusted/preferred); (3) *Design and implement the experiment*: the survey included a list of 12 wireless network names is compiled based on: they exist in the region where the study was conducted, non-existing, evocative of security or freeness and location/context-specific. The participants should be randomly associated with either the first or the second group of questions (between subjects design); (4) *Run the experiment*: send an email to the staff of the University of Luxembourg; (5) *Analysis*: Data is collected through an online survey, then analysed using R statistical tool. Basic descriptive statistics were applied followed by t-test and wilcoxon rank test.

We actually run such an experiment and report on parts of it in this thesis (the full details are to be found in [68]). Details on the influence of context are presented in Section 3.3 and used to validate STEAL’s support for social sciences methods, and the relevance of the addition of the context to STEAL’s reference model; details on the influence of trust are presented in Section 4.3.1.1 to illustrate the use of STEAL (in Chapter 4).

*Main results*: The social analysis confirms the hypothesis that SSID names reminding the context influence choices, but when users are unaware, or have not been instructed to use the official SSIDs. However, the study refutes the hypothesis that users trust names recalling “security”.

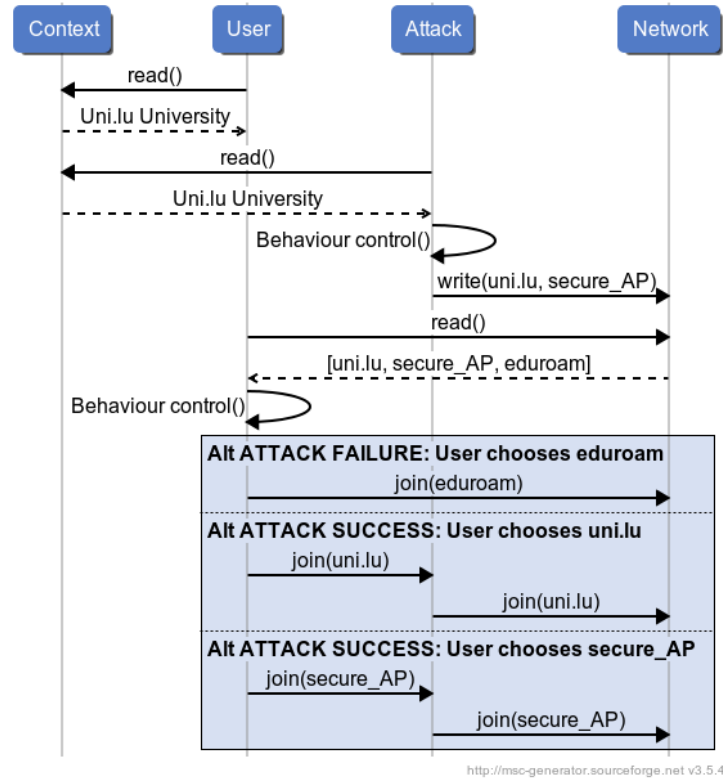


Figure 3.4: Technical Focus: the UML sequence diagram for the WiFi connection to the Internet with an intruder attacking the network.

### 3.2.2.3 Adding defences

After having identified possible attacks, we may devise possible defences. We sketch some of them in the remainder of this paragraph. Defences can act at the technical layers or at the social layers. For example, if all wireless access points were strongly authenticated by the user’s device, then the identified attack would not occur. This is likely the case with the new Hotspot 2.0, where the device’s SIM card embeds the certificate of proprietary access points. The network manager,  $p_A$ , can be programmed to disable the ‘join()’ action on all networks that have not been vouched by the university’s system administrator ( $\Theta_4$ , Figure 3.3). Another technical defence can be implemented at the Network layer by monitoring the live SSIDs, and spot whether some new SSID is trying to use the name of the context (e.g., the “uni.lu” SSID). Technically it is possible to disrupt the joining process to newcomer SSID by sending spoofed deauth packets. This action has the effect of disabling the ‘join()’ function ( $\Theta_5$ ). This defence is sometimes labelled “Rogue Containment” and is implemented in number of wireless network products (see [36] for instance).

If no technical solution is feasible, defences can be applied at the social layers or to the context. For example, stickers can be left all over the University campus, advertising the legitimate access point of the University ( $\Theta_1$ ). This may likely increase user’s awareness. The University can give training to its employees to help them recognise rogue SSIDs ( $\Theta_2$ ). The network manager and the user interface can have a trust indicator displayed aside each SSID ( $\Theta_3$ ).

Whether these defences are effective in successfully decreasing the number of people that fall victim of the attack herein described, is a research question that should be tested by new runs of our framework.

### 3.2.3 Summary

STEAL comprises a model of a socio-technical scenario and suggests methodologies to analyse and test the same scenario for its security. It helps modelling socio-technical attack scenarios too. At the moment, the methodologies for both technical and social security analysis are working in a pipeline, and allows more runs of analysis. The technical analysis justifies the presence of technical attacks, and the social analysis give ground to evaluate the effectiveness when user's decisions are in place with those attacks. The technical analysis cannot, at the moment, help with attacks of purely social nature, because there is no model able to express and simulate them. The same relates to mature human behaviour: there are no stable human behavioural models that can be used within an automatic security validation tool.

It is not a primary goal of STEAL to build a model for understanding why users behave the way they do. However, it is possible to use STEAL to design and perform experiments that focus on understanding why some users fall victims of a specific socio-technical attack, by following some behavioural patterns. Such findings may inspire defences, whose effectiveness can be tested in STEAL.

To validate STEAL reference model, operational guideline, and methodologies, we now study two use cases. The first one (see Section 3.3) aims at proving that STEAL's reference model is fine-grained enough to describe scenarios where an attack stems from the interplay of different actors, and that social sciences' methodologies can be successfully instantiated to investigate hypothesis regarding the success of such attacks. As we are in particular interested into the validity of the addition of the context into the reference model, the use case focuses on the role of the context into the selection of Wi-Fi networks. The second use case aims at showing that technical analyses of socio-technical systems can be performed through the use of model checking.

## 3.3 STEAL validation: social focus

In this section, we study STEAL's support of analysis focused on the social layers of a system (see Figure 3.5). In particular, we investigate if STEAL's reference model is fine-grained enough to be used to describe attacks that stem from the interactions between the attacker, the context, and the user.

Furthermore, we challenge the addition of the context's relevance into STEAL's reference model. The following study intends to prove experimentally that the context indeed explains part of the user's insecure behaviour in the running example used previously (see Figure 3.3) and that the presence of the context in STEAL's reference model is justified.

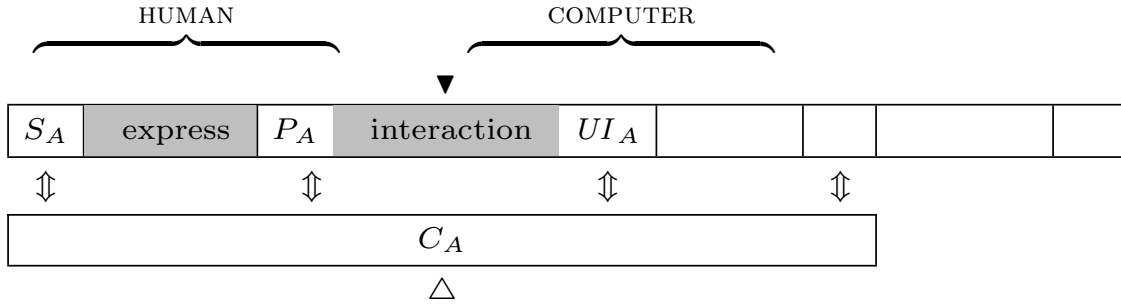


Figure 3.5: The multi-layered security and threat model used in this section.

### 3.3.1 Introducing the study

User do not know what behaviour to expect from a network before interacting with it. For instance, a Wi-Fi hotspot can provide access to the Internet immediately after the user has joined the network or it can redirect the user to a captive portal. To add further confusion, captive portals have no pre-defined behaviours before they provide access to the Internet; some only ask for the acceptance of conditions of use, some ask the user to enter personal information, and others require the payment of a fee or the provision of credentials. In these initial interactions, even tech-savvy users that protect their privacy by using end-to-end encryption while using open Wi-Fi networks can be at risk because before accessing the Internet end-to-end encryption is not active yet (unless users bypass the network’s restrictions by tunneling IP over DNS for instance, but this is a technical action that not all users can take.)

In this setting, users have to rely on their knowledge and beliefs to define various heuristics to select a network that appears to fit their expectations. We stipulate that the SSID and its relation with the surrounding context may be factors in the user’s choices, and that understanding how these factors influence the user’s choices can improve the effective security of Wi-Fi networks.

We imagine the mind-set of an attacker who intends to set-up a fake WiFi access point and who speculates on the best strategy to name it to “phish” people. A good strategy could be to choose names that relate to the context.

Context, is the physical or the social space where actions and decisions occur (e.g., in a laboratory, at work, at home). By addressing context we are interested in understanding whether this factor has an effect on people’s choices of names. If that is true, an attacker can be more effective by contextualising his/her attack or by fooling users to be in a context favourable to him/her. However, this brings new ideas on how to contain these context-exploiting attacks, for example by securing the access to the context (cf. Section 3.3.5).

In summary, the aim of this section is to present a study that investigates the effect that the context has on users when choosing wireless network names. Our study relates to decisions that do not require complex probabilities, balancing risks, or evaluating security with respect to goals: in such complex scenarios, user choices are ruled by principles of mental economics [9, 3], out-of-scope here.

This study also investigates the role of trust in the selection of Wi-Fi names, but we don’t report these results in the present section to focus on the relevance of the



Table 3.1: Existing/nonexistent wireless names and their grouping in relation to security and context. Security: (G1-existing; G2-nonexistent; G3-nonexistent and related to security; G4-nonexistent and not related to security). Context: (L1-existing and expected in the context; L2-existing and not expected in the context; L3-nonexistent and expected in the context; L4-nonexistent and not expected in the context).

	Security				University				City Center				Shopping				Hospital				
	G1	G2	G3	G4	L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4	
eduroam	■				■				■					■				■			
uni-visitor	■				■					■					■				■		
uni-student		■		■			■								■						■
wifi_unilu		■		■			■								■						■
hotcity	■					■			■					■				■			
Hotel_le_Place_D'Armes	■					■			■					■				■			
Cafe_de_Paris	■					■			■					■				■			
secured_hotspot		■	■	■				■				■			■						■
secure_wifi_BelleEtoile		■						■						■							■
free_wifi_BelleEtoile		■						■						■							■
Maroquinerie_Kirchberg		■						■						■							■
free_AP		■		■				■						■							■

context in STEAL’s reference model. The investigations on user’s trust are detailed the next chapter.

### 3.3.2 Use Case Scenario

Our hypothetical use case scenario consists of a set of wireless network names (SSIDs), various locations, and a user. The user is expected to scan and choose an SSID from a list of names that his/her device detects to get Internet access. This can happen in four different well known locations: the university, a shopping mall (a specific one), the city center, and a hospital (a specific one).

On the other hand, our scenario imagines an attacker whose intent is to deploy a dishonest WiFi base station. This station’s name will appear in the list of available SSIDs that the user can browse from its device. The attacker seeks to maximise the number of victims, so s/he looks for alluring names that takes advantage of the location to inspire legitimacy with names such as ‘wifi\_unilu’. Table 3.1 shows a comprehensive view of the 12 SSIDs used in this study, including those existing and those made up. The SSIDs have been carefully compiled: they may or may not exist in the region where the study was conducted, evoke security or free use, or be location-specific.

**Research Question.** The research question we intend to answer regarding preferences in wireless network names is (Context\_RQ): *Does context affect participants’ preferences?*

### 3.3.3 The Survey

For reasons of feasibility and ethics we opted for a survey rather than an experimental setup, the latter being e.g., the setup of a ‘malicious’ access point airing different SSIDs. Our survey asks respondents to rate a list of SSIDs according to their preferences while excluding technical aspects such as signal strength or protected access. Our survey relies on an online questionnaire rather than a paper-pencil version that would have required a large logistical effort to field and to encode, while not offering the same level of convenience to the respondent. The questionnaire was structured into three parts: (1) the socio-demographics part that surveys respondents about their age, gender, education, IT skills and comfort using IT; (2) the ‘general preferences’ part that lists 12 SSIDs the respondents are asked to rate with regard to their general preferences based on a 5 point Likert scale (i.e., 1-Not at all preferred, 2-Not very preferred, 3-Neutral, 4-Preferred, 5-Most preferred), respectively; (3) the ‘context’ part consists of 4 specific and familiar locations, each of these locations listing the same 12 SSIDs, asking respondents to rate them regarding specific contexts when connecting/avoiding them (same Likert scale as for the general preferences).

The instructions provided to the respondents have been translated from English to German and French in order to accommodate the multilingual population of Luxembourg and surrounding areas. As mentioned earlier, a research question about trust is also investigated in this survey, so, in reality the respondents were randomly associated with one of two conditions. From the point of view of the participant assigned to the context condition the survey was presented as follows: socio-demographic  $\rightarrow$  general preference  $\rightarrow$  context. We recruited participants by sending an invitation via email to students and staff from the University of Luxembourg.

Data were collected within a MySQL database and exported to a CSV file format. Statistical analyses were done using the R statistical analysis software [144]. The collected data were analysed using basic descriptive statistics, followed by specific analysis of variance tests (t-tests [124] and Wilcoxon rank [186] tests) in order to assess the significant differences and between general preferences and the context. In order to apply t-tests on data derived from Likert scales, we systematically verified its normal distribution and also employed the Wilcoxon signed-rank test to further support t-test results. We also included open questions (analysed manually) that allowed respondents to provide the rationale for their ratings.

A total of 235 participants took part in our study; however our analysis focuses on the 99 completed cases (136 cases have not been fully completed and thus have not been considered for analysis), and 40 participants were assigned to the *context* condition. As shown in Table 3.2 our sample is rather balanced with regard to gender. On average our respondents are rather young (age 26), mostly highly educated (over 75% have a bachelor degree or higher), very IT literate and highly skilled (75%).

Next, we present the results obtained for the *trust* condition. Whenever possible, we proceed by first describing general tendencies as visualized through graphical representations, followed by more specific analyses whose results are presented as tables. Differences between repeated measures have systematically been computed as follows: measure 2 – measure 1. Negative differences suggest than on average

Table 3.2: Sociodemographics for the population of the survey for the *context* condition, and to whole survey

Demographics	Condition <i>context</i> (n = 40)	Total (n = 99)
Female	58%	<b>45%</b>
Male	42%	<b>55%</b>
Age (average)	25%	<b>26%</b>
High School	28%	<b>22%</b>
Bachelor Degree	50%	<b>49%</b>
Master Degree	7%	<b>15%</b>
PhD	13%	<b>11%</b>
Very comfortable using IT	73%	<b>70%</b>
Somewhat comfortable using IT	25%	<b>26%</b>
Very good IT skills	23%	<b>29%</b>
Good IT skills	60%	<b>46%</b>
Average IT skills	15%	<b>21%</b>

Table 3.3: Statistical significance for the differences between general preferences and the contexts (in this case, there is no statistical significance for the context “city center”).

	Difference (Context preference-generic preference)		
	University	Shopping Mall	Hospital
Whole sample	-0.15* <sup>#</sup>	-	-
Females	-	-0.23* <sup>#</sup>	-0.33* <sup>#</sup>
> 24 years old	-	-	-0.27* <sup>#</sup>
> bachelor degree	-	-0.32*	-0.37* <sup>#</sup>

**Legend:** For all tables superscripts have the following meaning: t-test result: \* $p < 0.05$ ; \*\* $p < 0.01$ ; \*\*\* $p < 0.001$ . Wilcoxon result: <sup>#</sup> $p < 0.05$ ; <sup>#</sup> $p < 0.01$ ; <sup>#</sup><sup>#</sup> $p < 0.001$ .

measure 1 > measure 2 and positive values suggest measure 2 > measure 1. More precisely, a negative value indicates a decrease in preferences and conversely a positive value suggests an increase in preference. The statistical tests inform us on the significance of these differences.

### 3.3.4 Context

Fig. 3.6 displays the SSID preference ratings for only 4 of the 12 names that show some change throughout the contexts (i.e., University, City Center, Shopping Mall and Hospital) as compared to the general and non-context dependent situation, which is labeled “generic” in the figure.

Table 3.3 shows the significant results about the effect that context awareness has on respondent’s names preference ratings.

Significant results in the context condition indicate a decrease in preference ratings when respondents are made aware of specific contexts. This applies to the University context where the effect is demonstrated for the entire sample of respondents and, only for specific sample groups in the shopping mall and hospital context.

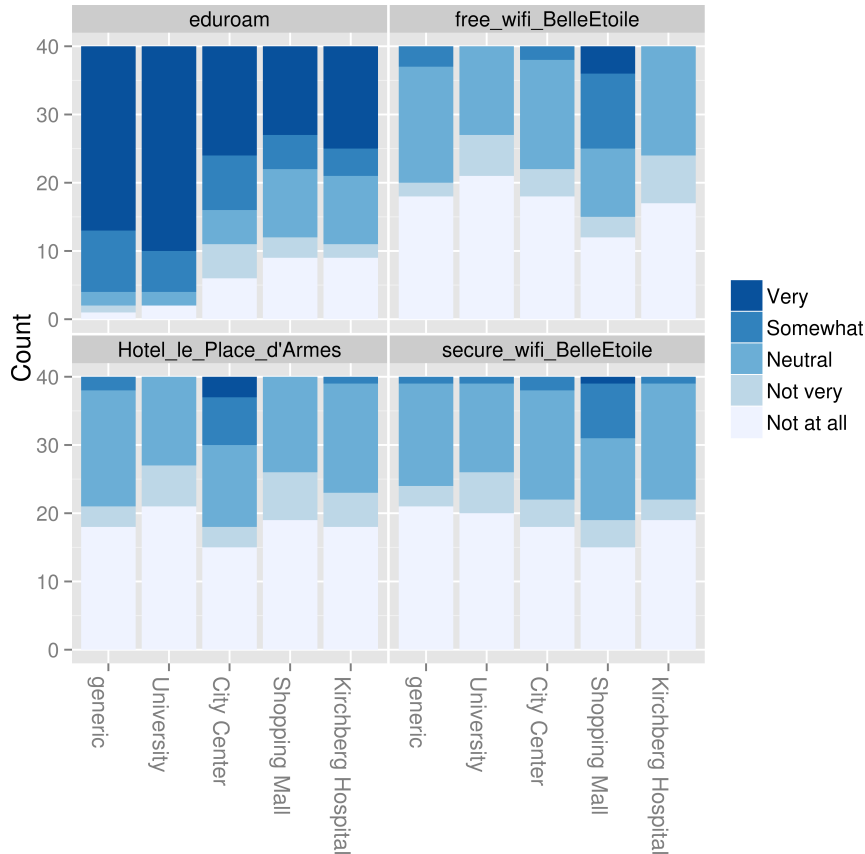


Figure 3.6: Selection made for eduroam, HotelLePlace\_d’Armes, secure\_wifi\_BelleEtoile and free-wifi\_BelleEtoile within the four contexts by all participants of condition 2.

The shopping mall indeed seems to demonstrate an effect specifically for female respondents and for those who are more educated. This is also true for the hospital context, the results indicate an effect for respondents aged more than 24 years old. These effects indicate that these respondents may be more aware when choosing a name for those three contexts.

We completed our analysis by a specific name grouping, illustrated in Table 3.1.

Fig. 3.7 compares between general preferences and the four groups (L1-L4) for all the contexts. Participants rate higher the SSIDs for L1 - existing and are expected within the university and the city center while in the other two contexts (e.g., shopping mall and the hospital) participants rate higher the names for L2 - existing but are not expected in that context. The figure also shows a tendency for participants to rate higher nonexistent wireless network names but which may be expected in the context (L3) (e.g., for the university, shopping mall and hospital contexts).

Table 3.4 provides an overview of the effects that the University context has on user’s preferences. Group L1 of “existing names and expected in the context”, are all affected by the university context in the sense that these names are rated higher, respondents thus being more cautious when context-aware. In contrast, group L2 of “existing names but not expected in the context”, have been rated lower when awareness about the context was included, except for male respondents.

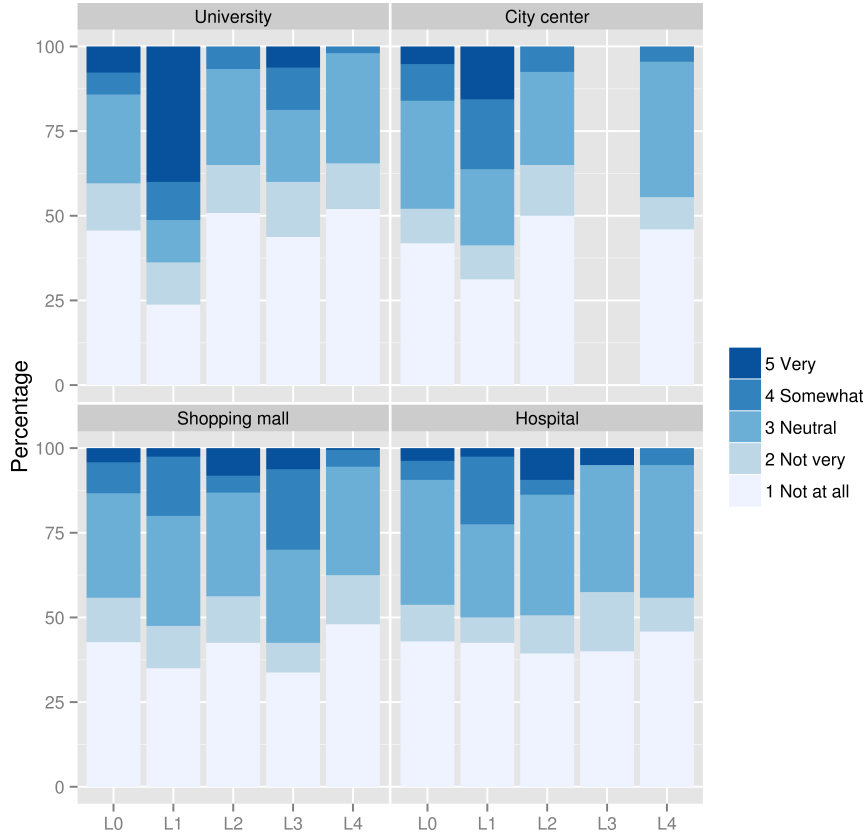


Figure 3.7: General preferences in the 4 groups (L1-L4) for all the contexts.

Table 3.4: Statistical significance for the differences between general preferences and the context of the University.

	Differences (L* - generic)			
	L1	L2	L3	L4
Whole sample	1.00*** <sup>##</sup>	-0.40*** <sup>##</sup>	-0.10 <sup>#</sup>	-0.47*** <sup>##</sup>
Males	0.86*** <sup>##</sup>	-	-	-0.48*** <sup>##</sup>
Females	1.10*** <sup>##</sup>	-0.50*** <sup>##</sup>	-	-0.46*** <sup>##</sup>
≤ 24 years old	0.99*** <sup>##</sup>	-0.37*** <sup>##</sup>	-0.22 <sup>#</sup>	-0.43*** <sup>##</sup>
> 24 years old	1.03*** <sup>##</sup>	-0.47*** <sup>##</sup>	-	-0.55*** <sup>##</sup>
≤ Bachelor Degree	1.01*** <sup>##</sup>	-0.35*** <sup>##</sup>	-0.18 <sup>#</sup>	-0.40*** <sup>##</sup>
> Bachelor Degree	0.95*** <sup>##</sup>	-0.60*** <sup>##</sup>	-	-0.71*** <sup>##</sup>
≤ Good IT skills	1.02*** <sup>##</sup>	-1.41*** <sup>##</sup>	-	-0.46*** <sup>##</sup>
> Good IT skills	0.94*** <sup>##</sup>	-0.39*** <sup>##</sup>	-0.22*** <sup>##</sup>	-0.50*** <sup>##</sup>

Table 3.5: Statistical significance for the differences between general preferences and the context for: (a) Shopping Mall, and (b) the Hospital.

	Differences (L* – generic )			Differences (L* – generic )		
	L3	L4		L1	L3	L4
Whole sample	0.29*	-0.36***#	Whole sample	-	-0.19#	-0.28*#
Male	0.43*#	-0.48***##	Males	0.69*	-	-
Female			Females	-0.49***#	-	-0.44***##
≤ 24 years old	0.43***#	-0.32*#	≤ 24 years old	-	-	-0.22#
> 24 years old	-	-0.44***##	> 24 years old	-	-	-0.40***##
≤ Bachelor Degree	0.38***#	-0.30*#	≤ Bachelor Degree	-	-	-0.22#
> Bachelor Degree	-	-0.56***#	> Bachelor Degree	-	-	-0.49***#
≤ Good IT skills	0.40***#	-0.43***##	≤ Good IT skills	-	-	-0.32*#

(a)

(b)

Table 3.6: Most common reasons for general preferences and each context.

	General pref.	University	City Center	Shopping Mall	Hospital
Do not use other networks	34	11	7	3	2
Do not know other networks	15	8	7	-	-
Easy Access	9	7	5	2	3
Security	5	1	3	3	1
Place where I am	-	-	9	10	2

The “nonexistent and not expected names in the context” (L4) have systematically been rated lower. Finally, the “nonexistent and expected” names (L3) show a weaker effect on the entire sample and higher effects for subgroups of respondents younger than 24 years, with less than a bachelor degree, or proficient with IT.

Table 3.5.(a) provides an overview of the effects that the shopping mall context has on user’s preferences. This context seems to be associated with a less pronounced effect on user response patterns as there is no significant difference for groups L1 and L2. However, there is a series of effects indicating a rating increase in subgroup L3 and a general decrease in ratings for L4.

Table 3.5(b) provides an overview of the effects that the hospital context has on user’s preferences. This context is associated with few significant effects. Results for L1 indicate positive ratings for males while the opposite for female respondents. There is also a decrease in ratings for the whole respondent sample in L3. And finally, consistent with results in Table 3.5(a), L4 names are systematically rated lower, except for male respondents.

Table 3.6 shows the results for the open questions relating to context. Again, the most common reasons relate to the use and knowledge of the network names, and that they provide easy access. To note that outside the University context, the most common reason states clearly that the place where the participants are, can greatly influence their choices.

### 3.3.5 Security Discussion

In our scenario the attacker pondered the best strategy for naming his malicious SSID to “hook” the most people to choose it when accessing the Internet.

Fig. 3.6 shows that people prefer a network that communicates a context-specific meaning. For example, the made-up ‘free\_wifi\_BelleEtoile’ rated higher in the shopping mall context than in general (Belle Etoile is an existing shopping mall, where there is no existing SSID reminding of that name). This can appear obvious, but Fig. 3.7, which shows the results for groups gives more useful insights. In the context “Shopping Mall” the increment is positive for all the made-up networks that refer to it (e.g., cf. Table 3.5.(a) first row, first column); but in context “University” this does not happen. Here, made-up names referring to the context (group *L3*, which includes ‘wifi\_unilu’ for example) rated less on average (cf. Table 3.4 first row, third column)<sup>1</sup>.

Our sample, mostly students and employees of the university, know better what network is available at the university. They do not expect networks to appear without notice. Thus, the strategy of contextualizing names has less impact at the university, at least for the possible victims who regularly frequent the university, as our population. However, it may work for guests or visitors, who may not be so aware of what access point exists.

In fact, in contexts like the shopping mall, the same strategy of contextualizing made-up names works nicely: those names out-rate the existing ones. An attacker targeting public places can thus increase odds by including the context in the name of a dishonest base station. Conferences, for example, are sites where such an attack could work very well.

What could be a recommendation to prevent such kinds of attacks? One suggestion, which could be tested for efficacy, would be to advertise the names of legitimate networks, for example by deploying stickers informing visitors about the legitimate access points. (An attacker can do the same, but this requires him to work and expose himself more). Another defence consists in avoiding to leave unused names which are related to the context. For example, a hotel should re-name SSID with the hotel’s name. Such simple action is usually disregarded: it is common to see WiFi with the name of the router (e.g., ‘linksys01’) or with that of the network provider (e.g., ‘Numericable\_6A85’).

### 3.3.6 Conclusion

Our result shows that, in unfamiliar contexts the choice of even expert people is biased towards names reminding of the context.

These results suggest severe socio-technical attacks that can be easily launched by interfering with user’s knowledge of the context. Furthermore, these finding shows that STEAL’s reference model is relevant for studying the social layers of security analysis. Indeed, the socio-technical analysis of this system would not be

<sup>1</sup>We got a similar despite weaker result for the context “Hospital” but with a different explanation. The contextualized name ‘maroquinerie\_Kirchberg’ is ambiguous because Kirchberg is also the name of a large zone of the city where the hospital and many other offices stand, while Maroquinerie is out-of-context.

possible without the addition of the context, as part of the explanation for user’s insecure behaviours would stay out of reach.

The study carried on in this section has some limitations. We did not have a larger and more diversified population, as we had permission to broadcast our survey only within the university. The small sample size did not allow for more complex multivariate statistical analyses. Also, not many participants filled the open questions. We think it would also be useful to analyse in more detail each wireless network name separately and verify its statistical significance. It may be that one or two names have more meaning than others and can in themselves be used to improve or mitigate socio-technical attacks.

We would have liked to set up attacks with real WiFi access points in real places; however launching such actions and harvesting the data for the analysis requires an authorisation from an ethical committee and a compliance with our legal framework, assurances that were not ready for this study.

Following this use case, we can conclude that STEAL’s reference model, and its inclusion of a context component is crucial in the study of the interactions that occur between the users, the Wi-Fi networks, and the context. Furthermore, the full social analysis was completed as an hypothesis was successfully expressed, an experiment successfully designed and ran, and conclusive results regarding the socio-technical security of Wi-Fi networks obtained.

### 3.4 STEAL validation: technical focus

In this section, we position our analysis on the technical layers (see Figure 3.8) to verify that STEAL allows for the use of computer sciences methods and that their outputs can be used as inputs for investigating the further the social layers (see Figure 3.2). We aim in particular at showing how the reference model can be used to describe the technical aspects of the interactions between a system and its users, and how a model of these interactions can be checked for subtle flaws.

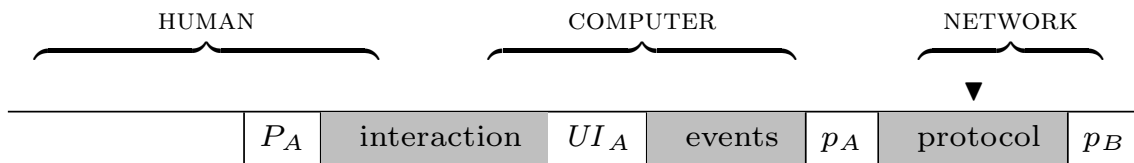


Figure 3.8: The multi-layered security and threat model used in this section. There is no context. The intruder is a man-in-the-middle represented by the inverted triangle.

For this investigation, we part from the Wi-Fi selection use case used previously. Indeed, Wi-Fi selection is too trivial for needing a technical study that visits all traces within possible interactions. Instead, we propose to study the authentication of identities with TLS certificates in modern Web Browsers. The diversity of implementations of TLS certificates’ validation calls for the use of verification methods to find subtle cases where the browser-user interaction threatens security.

We study how STEAL supports the technical identification of insecure interactions between users and Web Browsers. First we introduce the use case, then



we study the interactions between the technical and the social layers to identify problematic interaction traces.

### 3.4.1 Introduction

Let us assume you are meeting a person that is expected to do some job for you. You know nothing about him except a few things such as his name and affiliation. How can you trust a stranger introducing himself and claiming to be that person?

This “are you a friend or a foe?” problem boils down to assessing the validity of an identity. Plenty of solutions have been proposed to solve it; a few require sharing passwords beforehand (which is feasible only rarely) and the majority requiring a certain degree of trust. For example, trust is required when the stranger shows a document vouching his identity, because you need to trust the authority that has issued the document you are given. This problem of assessing someone’s identity is exactly that which, on the Internet, a browser faces when it asks a server to authenticate itself. Here, the proof has the form of a Transport Layer Security (TLS) certificate, which, to be valid, should be ultimately signed by an authority that the browser recognises as trustworthy. Indeed, if all certificates were signed by world-wide known and unquestionably honest authorities the role of trust would be negligible: no trust is required when there is full knowledge and control of events. Such a situation is unrealistic for TLS-based authentication (e.g., see [107]), and therefore, someone, somewhere, has to decide whether to trust an entity to be honest or not. But users do not usually have the understanding and security knowledge to take secure decisions. In TLS certificate validation users can be asked to take such decisions, for instance, when a server presents a self-signed certificate.

A self-signed certificate is issued and signed by the server itself. Thus, to trust a self-signed certificate one should trust the server already, which is a useless circular reasoning. What solutions are available that help security with self-signed TLS? The protocol that is responsible for the validation of certificates does not give answers: for TLS, self-signed certificates are technically unverifiable and thus “*MUST either notify the user (clients MAY give the user the opportunity to continue with the connection in any case) or terminate the connections*” [148]. This recommendation has been picked up by browsers that implement it in various ways; but since so many Internet frauds still happen, a reliable standard solution seems yet to be found.

Fraud can happen when users are involved in security decisions as, these can be very complex to explain and understand. Preferably users should better not take such decisions, as suggested in [42], [50], and [30], but for the problem of validating an identity on the Internet this seems not possible: in particular the question whether to trust a self-signed certificate is a process that is inherently socio-technical, for it is made of interactions between users, user interfaces, browsers, and even servers. Its security should be analysed by considering all those components and their interaction.

TLS technical security has been intensively studied (e.g., see [108, 77]), as well its usability (e.g., [151, 58, 165])) but until STEAL, there was no framework for a combined socio-technical security analysis.

We now apply STEAL, on a relevant problem with TLS certificate validation and self-signed certificates: the analysis of the interaction ceremonies between users and the most famous browsers in the market. The next Section describes the security analysis applied from the “Computer” to the “Network” layer.

### 3.4.2 Study

From Section 3.4.1 we recall that a necessary condition for a server’s identity to be authenticated is that the browser verifies the server’s TLS certificate. If it cannot, because, for example, the certificate is self signed, the success of authentication may depend on the user: a browser can ask him to decide whether to proceed or abort the session. Thus, TLS certificate validation is a socio-technical procedure made of communicating processes (the browser engine and the server), user interfaces (the browser’s window and the options offered thereon), and a persona (the user).

In this study we analysed four of the most popular browsers – Chrome, Firefox, Internet Explorer, and Opera Mini – and how they interact with users when they encounter a self-signed certificate. Since the four browsers run different engines (i.e.,  $p_A$ ) and ceremonies with users (i.e., interactions between  $P_A$  and  $UI_A$ ), the analysis of the structure of the dialogue browser-user is rich in possibilities.

This analysis (cf. Figure 3.8) is about the layers that span from  $p_B$  (server) to  $P_A$  (user). In fact, we modelled a server, a browser, an abstraction of the interface, and a simple model of a user who chooses non-deterministically among the options that are offered to him. Context is not necessary here, for the simple model of users is context-independent.

We tried different formalisms to model the entities in agreement with our multi-layered security model. We first used flow charts but, although intuitive, this formalism was not the best to model multi-layered interactions, aside from the fact that they lack formal semantics, a limitation that precludes any formal analysis. We thus chose UML activity diagrams, a fortunate choice for three reasons. They fit well with the layered representation, they give immediately an easy reading of TLS sessions (i.e., a quick glance at the diagrams of the browsers under study shows clearly their different validation mechanisms), and they can be easily translated into a formal language. We built the diagrams of Chrome and Firefox by looking at their official documentation and code. Internet Explorer and Opera Mini are closed-source, so we studied them empirically.

The pictures show a combination of mechanisms: Chrome complies with HTTP Strict Transport Security (HSTS) policy –a policy whereby a web server announces using only HTTPS–, Internet Explorer uses warnings extensively, Firefox is the most complete having a complex engine and elaborate user interactions, and Opera Mini clearly aims at being lightweight (as is designed for mobile platforms).

While modelling them in UML, we did not describe their full functionalities but limited them to how each browser treats certificate validation. Chrome diagram is shown in Figure 3.9). Our diagrams have four columns each representing communicating elements. Three are entities: User, Browser, and Server. Browser distinguishes two standard sub-entities: User Interface and Engine. Entities have a begin circle that points to their first activity. Thick arrows depict the flow of

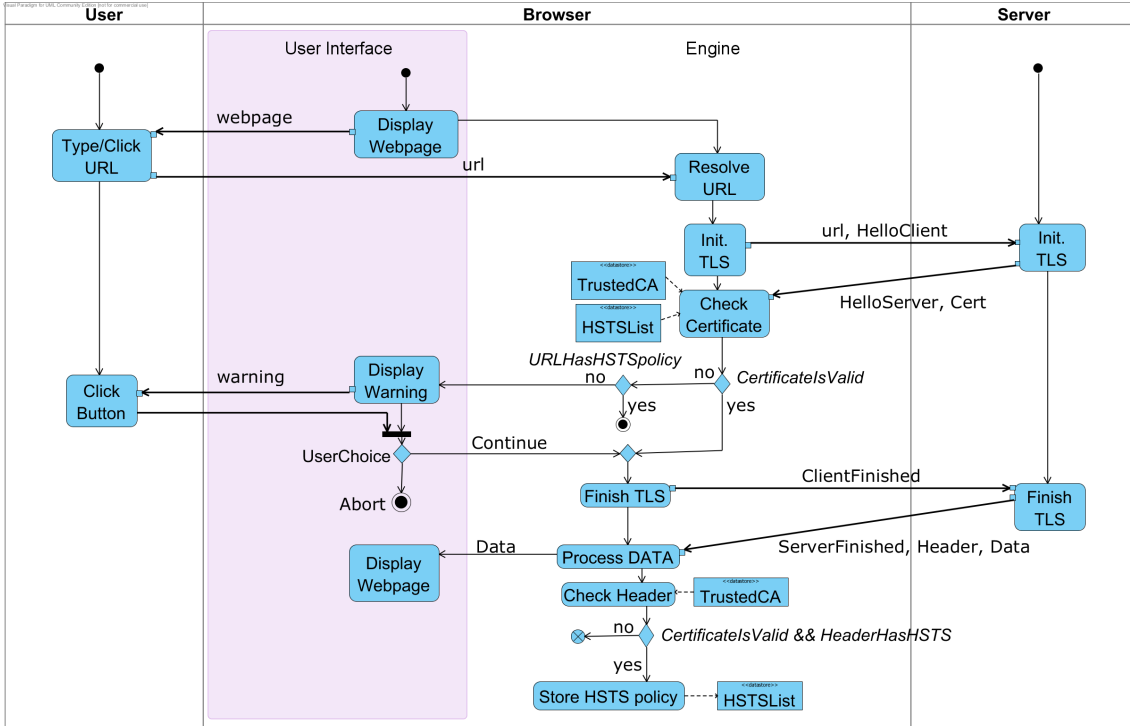


Figure 3.9: Activity diagram for TLS certificate validation in Chrome

activities among different entities, while thin arrows stand for the internal entity flow. Arrow labels define the objects that are exchanged between activities. Some activities need to access data-stores, which are linked to activities via dashed arrows. Most activities are self-explanatory and common to all browser diagrams, such as *Display Webpage* and *Type/Click URL*. To keep the focus on the browser, the server activities are reduced to *Init. TLS*, whereby the server starts the TLS handshake on its side, and *Finish TLS*, where it concludes the handshake. There is no room to describe in further detail the diagram, but it can be easily understood with a basic UML background.

To carry on the analysis we coded the UML diagram in a variant of CSP (Communicating Sequential Processes) [90] called CSP#; however, a prototyped tool that translates UML activity diagrams in CSP will be available soon [2], a tool we would like to test in the future. We also modelled an intruder, a Dolev-Yao controlling the network, and the user. Capturing the complexities of user behaviour by a formal model is a challenging open issue. As explained in the introduction, we modelled the user as a non-deterministic process: this is the weakest assumption about the user skills: a ceremony that is secure for a non-deterministic user, is also secure for any user.

The last step of our prototype methodology consisted in defining relevant security properties. We identified four socio-technical properties that bind TLS session, validation mechanisms, and user choices. We expressed them in linear temporal logic. One property is meant to evaluate the user involvement: it assesses whether the browser always warns the user when certificate validation fails. Two properties aim to evaluate whether the mechanisms that browsers adopt to manage failed

certificate validations protect users from man-in-the-middle (MIM) attacks (e.g., if browsers can prevent users accessing a page controlled by the intruder). The last property is about informing the user that a MIM attack might have occurred in previous TLS sessions.

We verified the properties with the PAT (Process Analysis Toolkit) model checker [164]. The most interesting results regard Firefox. PAT reports a trace showing that Firefox does not warn the user when a certificate validation fails. This is due to the drawbacks of storing server certificates permanently, which Firefox allows its users to do. Moreover, it is worth noting that no browser keeps records of past warnings, exposing users to vulnerabilities when they bootstrap with MIM. This finding suggests a novel, more secure, strategy for browsers. Browsers should maintain a cache of invalid certificate hashes. In doing so, it would be possible for browsers to warn users when a different invalid certificate is presented by a server with which the browser has communicated in the past. Looking into this strategy is a matter of future work.

### 3.4.3 Related Work

In this section we present some related work that has also focused on the security analysis and usability of browser security warnings and, in particular, of TLS certificates. There is much work available on the analysis of the TLS protocol but we only refer to two recent works which summarise most security problems. SSL/TLS certificate validation has many implementation vulnerabilities specially in e-commerce websites [77]. This section analyses SSL/TLS certificate validation in non-browser applications. In addition, little attention is paid to the problem of correctly authenticating the service provider by the users. Josang's et al. work [108] tries to develop a framework to provide for user authentication assurance. However, these researchers do not focus on the fact that a user authenticating the service provider will provide more ways to perform socio-technical attacks. Attackers will probably focus even more on social engineering to implement successful attacks. These authors' main conclusion is that it is essential to integrate user and server authentication in the same framework. So to devise a platform where both socio and technical aspects of TLS security can be developed and implemented together. We also share this view.

### 3.4.4 Conclusion

The tools selected to perform TLS certificate validation (i.e., UML, PAT, CSP) were adequate and helpful in expressing and evaluating in detail the security analysis and properties of the different browsers in study. STEAL's reference model helped us to identify, not only the events and protocols between computer and network components, but also the interactions between the human user ( $P_A$ ) and the user interface ( $UI_A$ ). With the obtained expressiveness it is easier and quicker to identify inconsistencies and vulnerabilities in the security properties which can be further corrected.

## 3.5 Discussion

After having introduced STEAL in Section 3.2 and illustrated its use for social analysis through the selection of Wi-Fi names in Section 3.3, and its use for technical analysis by the validation of TLS certificates in Section 3.4, we now discuss whether or not STEAL meets our expectations.

### 3.5.1 Reference Model

The first issue to be discussed concerns STEAL's reference model. We based STEAL's reference model on a Bella *et al.*'s *concertina model* [16] and added the concepts of attacks, defences and context.

Attacks and defences are needed in order to describe how the attacker strikes and how the analyst may attempt to thwart his attack. With the Wi-Fi selection use case, we used STEAL to investigate the role of the context and proved, by demonstrating that the context explains part of the user behaviour in the Wi-Fi selection, that the context is needed to describe STAs that make use of it. Investigating the social aspects of this interaction would have not been possible without the addition of the context to the framework's reference model. Consequently, the additions we made to the *concertina model* appear justified and allow to study attacks that rely on the complex interplay of the users their environment.

Furthermore, in Section 3.4, we used STEAL to describe the interactions that take place between Web-Browsers and their users while validating identities with TLS certificate, and we identified some subtle inconsistencies within these interactions that can undermine security.

### 3.5.2 Support to social and computer sciences methods for security analysis

As shown in the previous sections, one can analyse both technical and social layers of system's security through the lens of STEAL.

The study performed in Section 3.4 shows that technical security analysis can be applied generically once all components of a socio-technical system, together with its interactions, are modelled as suggested.

The study we report on in Section 3.3, successfully investigate an hypothesis related human decision of selecting a Wi-Fi network through the use of an on-line survey, which is commonly used in social sciences.

But, the main issue to consider is if STEAL can help to generalise this analysis for a large set of scenarios. We believe that the methodology used for the social analysis (hypothetico-deductive experimental model) is generic enough to be applied in the design and implementation of user related experiments for socio-technical systems. In order to perform the security analysis, all the steps of that process need to be clearly and objectively defined. It may be the case that we can only test one interaction, and therefore, one hypothesis at a time. Still, its analysis uses methods that either confirm or dismiss that hypothesis. Once we know this answer we can

step to the next question or generate some conclusion. This is still generic and prone to be adapted to different socio-technical scenarios.

Both use cases were chosen for their propensity to stress particular aspects of the framework but were not studied on the whole spectrum of analysis. Indeed, we did not use formal methods in the Wi-Fi use case for finding security-harming interactions being trivial, and we did not complement the TLS validation use case for the demonstration of a social analysis had already been done in the Wi-Fi use case. Therefore, the operational guideline that allows to use technical and social analyses in a pipeline as pictured in Figure 3.2 has not been tested. The next chapter fills this gap by describing in detail the identification of critical action point (where the user's action can harm a security property) in a system through a technical analysis of the system, and the investigation of research questions related to these points through on-line experiment. Albeit the next chapter demonstrating the use of the operational guideline in the next chapter, we complement the TLS use case with a social analysis in Annex A for the sake of completeness.

## 3.6 Conclusion

We presented a framework for analysing socio-technical security of systems, and applied it to study the role of the context in the selection of Wi-Fi networks, and the TLS certificate validation and self-signed certificates. The multi-layered security and threat analysis model presented in this chapter can express and integrate socio-technical interactions for all components within the system in analysis together with the context where that system is set. As is common knowledge, security can include several layers, and if we miss one of them, we can be opening vulnerability doors where those layers can be compromised. If we are able to see the whole picture and include the whole system (both human and technical) in the security analysis, it will help us find and tackle more vulnerabilities between all the involved elements.

Regarding the study of Wi-Fi names and context, the same multi-layered model helped us to identify where attacks and defences could be applied for both generic socio-technical attacks as well as socio-technical attacks for the described scenario. By focusing the security analysis within the interactions between  $(P_A)$  and  $(UI_A)$  and the context where they are set, a common attack was easy to define. In addition, some defence that can be used to tackle those attacks was also introduced. Of course this is still theoretical, but using the HCI research process where we base our socio-technical studies, this defence can be evaluated in practice with real user experiments. Again UML was useful to express this socio-technical security analysis.

An important contribution of STEAL is that it introduces the concept of context, which is many times ignored in security analysis as is usually hard to define. However, within socio-technical security analysis, the context is a very important component which cannot be ignored because it can influence the user's set of mind when he is taking security decisions [157]. The context is external to the system components' interactions but can interfere with all of them, and so is even more relevant to be included in system security analysis. For socio-technical vulnerabilities, attackers use many times the context where the system is set to influence or press the user into falling for their attack. In the same way, our research can be used

to find out if defences can similarly use the context to influence and help the user identifying and avoiding such attacks.

Regarding the study of the validation of TLS certificates, STEAL was useful to clearly model the flow of interaction between the different social and technical layers, and ultimately to find subtle inconsistencies that can render some security properties vulnerable. Thus, STEAL can help identifying where the system fails to interact properly with its users, dooming them to endanger security.

We believe that STEAL is a good first step in the integration of socio-technical security analysis by a multidisciplinary team. Nevertheless, there is the need to apply STEAL to model and analyse more socio-technical scenarios. Only this way will it be possible to improve STEAL and enrich its flexibility and generalisation.

In the next chapter we use STEAL on the whole spectrum of a use case, identifying in the use of Wi-Fi hotspots where security properties can be harmed by the users, and then testing hypothesis about the factors that could lead them to do so. We show how to identify points of interest in a system, and then how to draw and investigate research questions about why a user would harm the system's security at these points.





# 4

## Applying STEAL for the Socio-Technical security analysis of Wi-Fi hotspots

### Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>56</b>
<b>4.2</b>	<b>Raising Research Questions on critical actions</b>	<b>57</b>
4.2.1	Methods	57
4.2.2	Use cases	59
4.2.3	Informal Technical Security Analysis	63
4.2.4	Discussion	65
4.2.5	Conclusion	67
<b>4.3</b>	<b>Investigating Research Questions</b>	<b>67</b>
4.3.1	The influence of trust in the selection of Wi-Fi networks.	68
4.3.2	The influence of graphical cues in the selection of Wi-Fi networks.	73
<b>4.4</b>	<b>Conclusion</b>	<b>83</b>

---

*In this chapter, we perform a socio-technical security analysis of Wi-Fi hotspots' most salient security ceremonies. We start by identifying critical actions points, compiling potential attacks and research questions about the reasons of their success. Then, we investigate two of these research questions related to the selection of Wi-Fi networks: the first about the role of trust, and the second about the role of graphical cues. Part of the content of this chapter appears in three papers: "Socio-technical Security Analysis of Wireless Hotspots" by Ana Ferreira, Jean-Louis Huynen, Vincent Koenig and Gabriele Lenzini published in the Proceedings of the*

*second International Conference on Human Aspects of Security 2014, (HAS 2014, Heraklion, Crete, Greece) [66]; “Socio-Technical Study on the Effect of Trust and Context When Choosing WiFi Names” by Ana Ferreira, the author, Vincent Koenig, Gabriele Lenzini and Salvador Rivas published in the Proceedings of the 9th International Workshop on Security and Trust Management (STM 2013, Egham, UK) [68]; and “Do Graphical Cues Effectively Inform Users? - A Socio-Technical Security Study in Accessing Wifi Networks” published in the Proceedings of the third International Conference on Human Aspects of Security 2015, (HAS 2015, Los Angeles, CA, USA) [Best paper award] [69].*

## 4.1 Introduction

In this Chapter, we perform a socio-technical security analysis of Wi-Fi Hotspots with a user-centric approach. First, we use STEAL to informally analyse Wi-Fi Hotspots’ most salient security ceremonies and pinpoint the *critical action points*, or the crucial points where the user can harm a security property. Then we discuss the potential factors that could push the user to behave insecurely at these points, and what experiments could be conducted to investigate their influence. Eventually, we perform two online studies to investigate the role of two potential influencing factors in the selection of Wi-Fi networks: trust, and the graphical cues displayed with the network names to illustrate the networks characteristics.

The study of Wi-Fi Hotspots is interesting because they offer very diverse contexts of use, and modalities of access. Indeed, the increasing demand for WiFi Internet access is pushing several public spaces, such as hotels and airports, to offer Hotspots. These are open, unencrypted WiFi networks that may redirect mobile users to web sites where they have to pay a fee or accept some policy before being allowed to navigate the Internet. Hotspots are spreading fast for they are believed to be a solution to the overwhelming demand of high-bandwidth services which is presently saturating mobile networks. Unfortunately, current Hotspots offer little or no security [33][160], therefore Mobile Network Operators are hailing the newcomer Hotspot 2.0 [8]; this is expected to rely on a better technology [97], able to overcome present vulnerabilities by encrypting every interaction and isolating all client sessions.

Hotspot 2.0 main functionalities are twofold: (1) the seamless roaming enables Mobile Network Operators to steer some traffic off the 3G and 4G networks to WiFi networks without user intervention and (2) access points will be able to display information about their current load and available services before the user gains access to the network. The latter being surely useful for venues like a stadium facing very high demand in bandwidth due to some specific uses, like instant replays; the network could block unicast streaming traffic on the network and advertise the use of a multicast streaming service directly from the user’s connection manager [24]. Hotspot 2.0 is thus advertised as a progress, with better security and better user experience.

However, despite its superior technical security, the *effective security* of this new technology will depend on how people will make use of it. This aspect is crucial as

it has been proved that security mechanisms are rarely used by users as technically intended [114]. For instance, users may not trust Hotspot 2.0's new technology. Or users can accept it but the new acquired sense of security is no more justified if they switch back to conventional Hotspot, a situation that is possible since the old technology will continue to exist for some time, confusing users on what security risks can be present.

As introduced in Chapter 3, analysing security issues with people in the loop demands for a *socio-technical* approach. This implies to look at the technical and the human protocols and to consider them together as complex layered ceremonies [16][68][63]. There is no such study for Hotspot and Hotspot 2.0, neither comparatively nor separately.

**Outline.** This chapter covers this gap by describing Hotspot and Hotspot 2.0's most salient ceremonies and by studying their security with a user-centric approach. Its main goal is twofold: (1) raise future research questions and priorities about factors and mechanisms (e.g., user awareness, context, perception of security, trust) that may influence a more or less secure user behaviour in Hotspot's WiFi ceremonies, and (2) investigate two of the research questions identified in (1).

## 4.2 Raising Research Questions on critical actions

To devise research questions about the factors that can influence the user behaviour in Wi-Fi hotspots' ceremonies, we worked on four use cases that cover most of their diversity. In the following sections, we first model the use cases without any attacker (Section 4.2.2) and then perform a security analysis (Section 4.2.3). At the end of this section (Section 4.2.4), we outline the setup of experiments allowing to answer the research questions that have emerged throughout this study.

### 4.2.1 Methods

We use STEAL (see Chapter 3.2) to analyse the socio-technical security of each use case. We perform an informal technical security analysis by modeling the interactions between the different players of the ceremony with UML sequence diagrams and by systematically devising the possible attacks when these interactions are exposed to threats according to a pre-defined threat model.

**Modeling.** We model ceremonies with UML sequence diagrams, a formalism that was successfully applied in socio-technical security analysis of TLS certificates (see Section 3.4, and [19]; it visually expresses all the sequential interactions (both Human-Computer and Computer-Computer) run by the players in the ceremony. This modelling is crucial for it defines the sets of interactions that can be analysed individually, in group, or at different levels of inter-dependency.

In order to get an objective analysis of the different use cases, we divide the Hotspot ceremonies in common phases in which we identify one or more actions. Each action is the result of a decision, taken with or without user involvement.

*Prior* is the action that happens before the user enters the ceremony; this is an optional pre-requisite (e.g., getting a SIM card by mail for instance); *Entry* is the entry point of the user, where he performs his initial action (e.g., open a url); *Selection* is the phase where the wireless network to be used is chosen from the list of available networks; *Access* is the action needed to successfully connect to the Hotspot (e.g., pay a fee); *Use* is where the user will actually use the network (e.g., performs again the action he tried in the *Entry* phase).

**Informal Technical Security Analysis.** Our analysis takes the user's point of view in the possible presence of an attacker who interferes with the user at *critical decision* points. These *critical decision* points are decision points from which the user can lose data confidentiality and integrity if the attack succeeds. For example, sending sensitive data should only take place when the WiFi is honest or the communication is encrypted. But, at this given *critical decision* point (choosing to send or not sensitive data on a communication channel), the attacker may push the user towards the insecure behaviour, the *critical action* of sending the data. We first define the feasibility of the attacks through the following *threat model* and *assumptions*; then we identify the ceremonies' *critical actions* by assessing the user's risk in the security-analysis (Section 4.2.3).

**Threat model:** we consider two threats:

1. a Local Attacker (LA) that can read & write in the ether; it means in particular that it can bring up dishonest Access Points and listen to unencrypted messages;
2. a Distant Attacker (DA) that can read & write messages on the Internet; an attacker that provides a phishing link to the user falls in this category. LA and DA can also cooperate.

**Assumptions:** we have 2 assumptions:

1. we assume that all interactions are taking place during the *Prior* phase are honest,
2. we assume perfect encryption, meaning that the only way to decrypt encrypted information is by knowledge of the key. Under this assumption, HTTPS provides an unbreakable encryption and the honest server exposes a valid, verifiable certificate.

**Risk assessment:** the risk is described on a four-level scale:

*null* no attack is possible;

*low* the confidentiality or the integrity of user's action is threatened (e.g., when the attacker can listen to user's actions);

*medium* confidentiality of user’s data threatened (e.g., when the attacker can listen to user’s data);

*high* confidentiality and integrity of user’s data threatened (e.g., when the attacker can tamper with the user’s data).

**Critical actions:** are the actions for which the risk is at least medium, and also all other actions that are necessary for them to occur.

**Results:** we summarise the result of the analysis in tables. For each row – corresponding to a phase of the ceremony– we consider the following information in the columns:

*1st column* the *information* conveyed to the user,

*2nd column* the *actions* that the user can perform,

*3rd column* the *attacks associated* with this action,

*4th column* the *security property impacted* by these attacks,

*5th column* a graphical representation of of the resulting *risk* level.

The findings are further discussed in Section 4.2.4.

## 4.2.2 Use cases

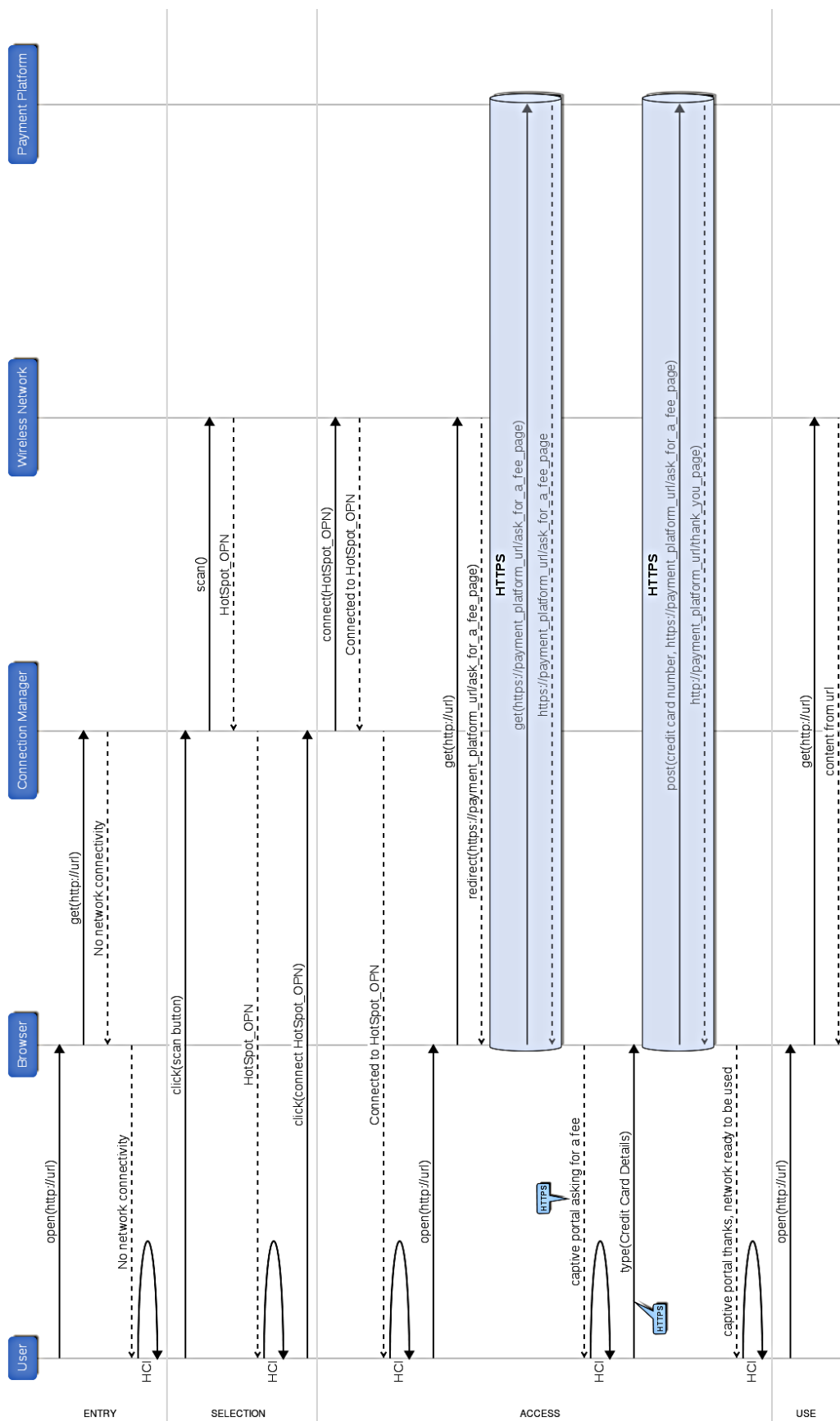
We choose four use cases that we think cover a large variety of situations. We concentrate on main differences like the automation (or lack of), the *selection* and *access* phases, the different types of actors (e.g., persons, service providers), the need to pay during the *access* phase, the changes made to the encryption over time, and the information load and quality. We only consider a few types of authentication for the sake of space.

The first two use cases relate to the Hotspot technology in use (abbreviated as HS1.1 and HS1.2) while the two last ones relate to the Hostpot technology users will encounter in the near future (abbreviated as HS2.1 and HS2.2).

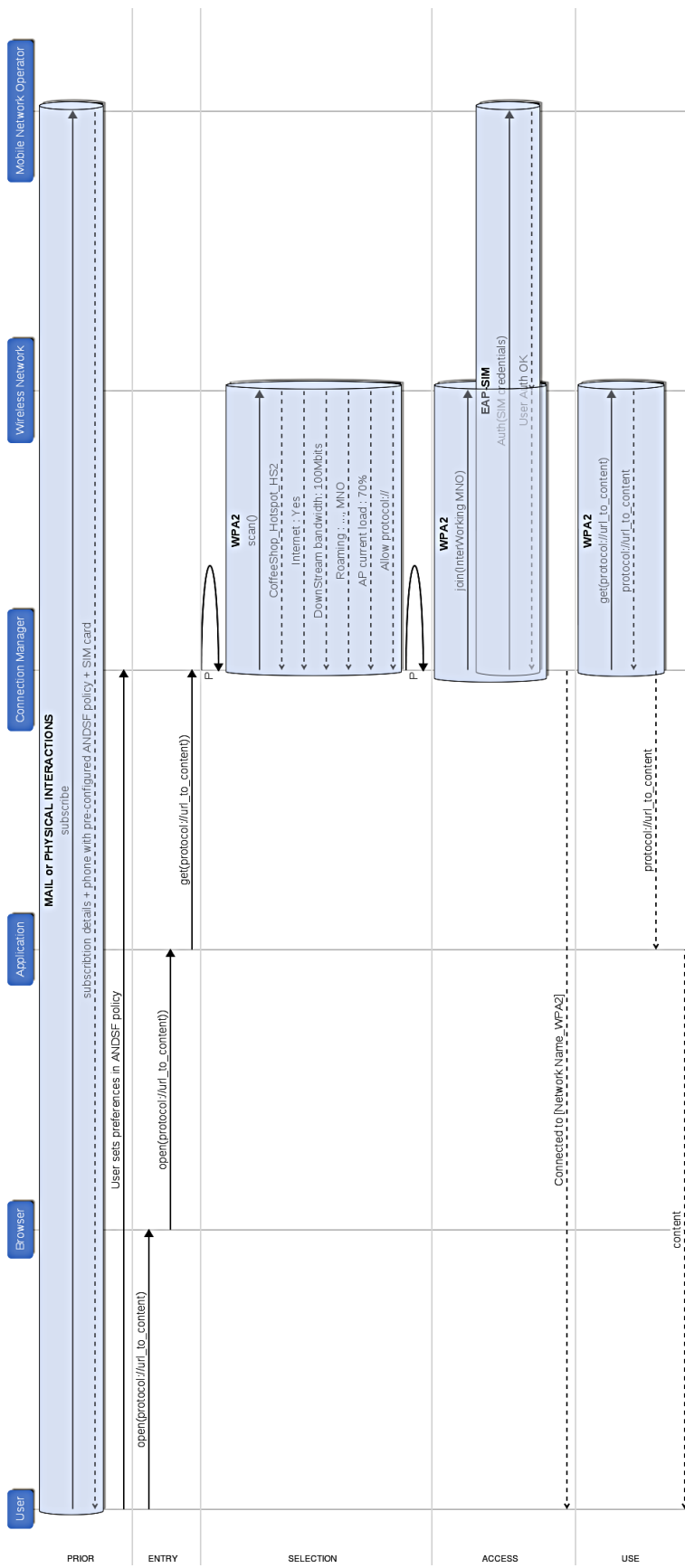
**HS1.1: Pay-per-use Hotspot.** Fig. 4.1a shows the UML diagram for the pay-per-use ceremony of a typical captive portal Hotspot<sup>1</sup>. The players are a user, a browser, a connection manager, a wireless network provider and a payment platform. The *entry* point is a user who wants to browse the Internet; lacking of Internet connectivity, he proceeds to the *selection* phase where he scans for available networks and connects to the pay-per-use unencrypted wireless network. In the *access* phase, the user is redirected to the payment platform to pay the fee. The browser runs an HTTPS session, which often carries the usual HTTPS browser cues (🔒), to execute the payment. After this step, the user is then free to *use* the (unencrypted) wireless network to browse the Internet. After having paid the fee, the security of the

---

<sup>1</sup>Captive portal : the user only has access to the Local Area Network until he pays a fee to be freed.



(a)



(b)

Figure 4.-1: (a) UML diagram depicting user’s interaction when joining a pay-per-use Hotspot. Components are at the top of each line, arrows represent exchanged messages: a plain line is used when a component initiates a message and a dotted line when a component replies to a message. The blue tunnels represent the messages communicated within an encrypted tunnel with HTTPS. (b) UML diagram depicting user’s interaction when Roaming on a Hotspot 2.0 Hotspot. Components are at the top of each line, arrows represent exchanged messages: a plain line is used when a component initiates a message and a dotted line when a component replies to a message. The blue tunnels represent the messages encrypted using WPA2 protocol or communicated within an encrypted tunnel with HTTPS.

underlying protocol decreases (e.g., HTTPS is no longer provided) and most common browsers do not effectively communicate this change to the user.

**HS1.2: Internet Service Provider’s Homespot.** This use case is what is commonly called a Homespot. This is a residential router provided by an Internet Service Provider (ISP) that reserves most of its bandwidth for the customer who owns the device, but offers part of its capacity to the passer-by customers. The players are the (passing-by customer) user, his device’s connection manager, the wireless network and the ISP. In the *prior* phase, the pre-requisites are that the user receives information (among these, the SSID) and his credentials. Using the same *entry* phase as HS1.1, the user then proceeds to the *selection* phase where he uses his connection manager to list the available Networks, and clicks on the one offered by the ISP. In the *access* phase, the browser is redirected to the ISP’s online website, over HTTPS, where the user enters his credentials. As these are valid, the user gets a feedback from the webpage that he is now free to *use* the (unencrypted) wireless network

**HS2.1: Mobile Network Operator’s partner Hotspot.** Fig. 4.1b shows the UML diagram for the ceremony of a user connecting to a Hotspot 2.0 through his/her mobile phone. This requires no user interactions except the *entry* phase as the device will follow a pre-defined policy called ANDSF [1] to decide what network to join, and will use its SIM card to authenticate to the Hotspot. The ANDSF policy comprises user’s preferences (e.g., always prefer user’s home network), the Mobile Network Operator (MNO) preferences (e.g., roaming partners), the application requirements (e.g., steering traffic from VOIP to WiFi) and the Hotspot’s conditions (e.g., the device should not switch to an overloaded Access Point). The pre-requisites (*prior* phase) are: the user gets the device pre-configured by his MNO, and sets some ANDSF preferences. The players are the user, the browser, an application, the connection manager, the wireless network and the MNO. In the *entry* phase the user opens a url in the browser which points to the content that requires the use of the application. The connection manager computes the policy bound to this application and concludes that it needs to connect to a WiFi wireless network. As a result, the connection manager automatically proceeds to the *selection* and *access* phases where it authenticates the user to the MNO. Once the connection is ready, the user is notified and, (*use* phase), the traffic corresponding to the content he



requested is steered to the wireless network (encrypted with WPA2 Enterprise). Eventually this content is displayed to the user in the corresponding application.

**HS2.2: The future of Hotspots.** This use case focuses on the cohabitation of conventional Hotspots with Hotspot 2.0 with services support<sup>2</sup>, when the automatic selection is disabled or impossible. The players are the user, the browser, the connection manager and the wireless network. The user's *entry* action is browsing the Internet; as there is no internet connectivity, he asks the connection manager to scan for available networks in the *selection* phase. The connection manager brings back results of: (1) conventional Hotspots with their SSID and signal strength; (2) Hotspot 2.0 networks with their SSID, signal strength, venue name, roaming partners, current load, WAN bandwidth, allowed ports; and eventually (3) services described by an icon and a url. The user then connects to one of the different candidates from the information at hand. Selecting (1) redirects the user to a use case like HS1.1; selecting (2) or (3) sends the user to the *access* phase where the network automatically provisions him an account. As a consequence, all following interactions are encrypted with WPA2 Enterprise and the connection manager notifies the user that he joined the network. The *use* phase is different for (2) and (3): in (2) the user browses the Internet, in (3) the user's browser is redirected to the url specified by the service.

### 4.2.3 Informal Technical Security Analysis

Our technical security analysis is user-centric, as such, its purpose is to pinpoint the critical actions prone to socio-technical attacks. Ultimately this leads to identifying upcoming research questions and possible laboratory experiments with users to be investigated in a subsequent social security analysis.

**HS1.1: Pay-per-use Hotspot.** Table 4.1 describes the security analysis of the HS1.1 use case. In the first phase of interaction the user scans for open networks. As this interaction is not encrypted, it can be eavesdropped by a Local Attacker (LA) so, according to our risk assessment procedure described in Section 4.2.1, the risk is set as low. In the *selection* phase, the user picks a dishonest network from the list. By this action, the attacker only knows that his network has been picked; the risk is low. The *access* phase is protected by HTTPS, which by assumption sets the risk to null. In the last phase, *use*, the user decides now to use the network, here the user can give away a lot of possibly valuable information to an eavesdropper and the attacker can even tamper with subsequent actions if the user formerly selected the attacker's network, so the risk is high. The *selection* and *use* phases comprise *critical action* points and will be further discussed in Section 4.2.4.

**HS1.2: Internet Service Provider's Homespot.** In the Homespot use case, the situation is closely related to HS1.1 as the user selects the attacker's network in the *selection* phase (again we set the risk as low). The attacker impersonates the ISP's wireless network but he can not (from assumption) tamper with the *access*

---

<sup>2</sup>We assume the use of the existing CISCO's implementation of Hotspot 2.0 services, called MSAP; see chapter 12 of [35] for additional information.

Table 4.1: Socio-technical security analysis of the classic pay as you go captive portal






Phase	Information	Actions	Associated Attacks	Security properties impacted	Risk
Entry	No connectivity.	scan()	Eavesdropping scanning action.	Confidentiality.	
Selection	List of available networks.	connect(dishonest)	Eavesdropping picking action.	Authentication of the AP.	
Access	Webpage asking for a fee. HTTPS cues.	enter(credit card details)	-	-	
Use	Network ready.	open(url)	Eavesdropping information. Tampering.	Confidentiality. Integrity.	

phase, as the connection to the ISP relies on the HTTPS protocol (the risk is null). The attacker lets the user authenticate to the ISP, like he would do on a legitimate Homespot. In the *use* phase, the user takes the decision to browse the Internet on this connection, similar to the previous use case. The risk is high as the user might lose confidentiality and integrity of his data. *Selection* and *use* comprise *critical* actions and will be discussed in Section 4.2.4.

**HS2.1: Mobile Network Operator’s partner Hotspot.** Table 4.2 describes the security analysis of this use case. In the *prior* phase, setting a ANDSF policy does not pose any risk. In the *entry* phase, opening a url is considered as low risk because a DA can write a url in the Internet that, when clicked by the user, triggers the network discovery. The *selection* phase’s actions are performed by the connection manager following the ANDSF policy (which has been altered by the user). The user can set a preference in the *prior* phase to rate unauthenticated, free Hotspot higher than the authenticated MNO’s partners; this can be exploited by a LA which would provide a Hotspot 2.0 with corresponding characteristics. The risk would be high as the LA could eavesdrop and tamper with the user’s data. LA and DA can also cooperate: LA can set an appealing hotspot while DA triggers network discovery. Both *critical actions*—setting a loose ANDSF policy and using a dishonest network—will be discussed in Section 4.2.4.

**HS2.2: The future of Hotspots.** This use case focuses on the *selection* phase when the automatic selection of a Hotspot is disabled. The user has to deal with different information emanating from different networks. The risk of connecting to a dishonest **network** that exposes appealing properties is high as it would lead the user to compromise his data confidentiality and integrity in the last phase of the ceremony. The risk of selecting a dishonest **service** is even worse as the user would be automatically redirected to the url set by the LA. The factors that can influence this critical decision will be discussed in Section 4.2.4.

Table 4.2: Socio-technical security analysis of an automatic roaming to a Hotspot 2.0 through an ANSDF policy

Phase	Information	Actions	Associated Attacks	Security properties impacted	Risk
Prior	SIM card. MNO information.	User sets its ANSDF preferences.	-	-	
Entry	url.	open(dishonest url)	Trigger Network Discovery.	Authentication of source action.	
Selection	-	-	Appealing Hotspot 2.0.	Authentication of AP.	
Access	-	-	-	-	
Use	Network Ready.	open(url)	Eavesdropping. Tampering.	Confidentiality. Data Integrity.	

#### 4.2.4 Discussion

For each *critical action* pointed out in the previous section, we elaborate on the following items: (a) research questions emerging from the *critical actions* about what factors (e.g., user’s perception of security and trust, or user’s awareness) affect the user’s critical decisions; (b) experiments that need to be conducted to answer these questions.

##### HS1.1: Pay-per-use Hotspot

*Selection phase: the user connects to a dishonest Hotspot* As the only information conveyed to the user at this point is a list of available WiFi networks along with the usual visual cues displayed by network managers, the research question is: (a) what is the influence of the context, the signal strength and the likeliness of the name on the user’s preferences? (b) In-vivo experiments based on deception (under strict compliance with ethical requirements like those of American Psychologists’ Association - APA) followed by a survey are relevant to assess the importance of these different factors. Surveys and laboratory experiments where participants would have to choose from a network list to fulfil a high-stake task are relevant to refine our findings. Also, contrasting self-reported behaviour (surveys) with observed behaviour (e.g. lab experiments) would be useful to investigate users’ awareness.

*Use phase: the user uses a dishonest Hotspot* As the user just pays a fee through an HTTPS connection before this *critical action*, we focus on the perceived changes of the security properties. (a) Are users aware that security properties change over the course of this ceremony and that after a successful payment, subsequent ceremonies are done in an open/unencrypted connection? If users are aware, what is their degree of awareness and how does that affect their subsequent actions? If users are not aware, do they feel the same sense of security during the whole ceremony or does it change at different stages? Do they perceive the signal and cues that can

trigger user awareness for the change? Is there any more adequate contextual information that could improve users' perception of this change? (b) The main challenge here is to investigate how HCI factors impact the awareness and responses to security properties. Laboratory experiments can be set up, e.g., using different security properties as different conditions ideally in a between subjects design. Comparing user behaviour across the conditions would provide strong indicators that could be further understood through interview techniques.

### **HS1.2: Internet Service Provider's Homespot**

*Use phase: the user uses a dishonest Homespot* We focus here on the impact of an unauthenticated and authenticated interaction with the ISP. (a) Does impersonating an ISP tend to foster a trust relationship with the network? Does interacting with the ISP through a secured-connection foster a trust relationship with the network? Is this true for any player representing authority? (b) Those questions can be investigated with laboratory experiments or online surveys: users would be placed in scenarios where they have to perform critical activities (e.g., e-banking) through different networks—some impersonating ISPs, some authenticated as ISPs. Comparing user behaviour across these different conditions provides indicators that could be further understood through interviewing techniques. One important aspect in these experiments consists in reliably simulating the “risk” without compromising ethical requirements.

### **HS2.1: Mobile Network Operator's partner Hotspot**

*Prior phase: the user sets a loose ANDSF policy* This decision can be linked to economic considerations, as the MNOs will sign many roaming agreements with different partners, they may keep track of the amount of data consumed by their customers when roaming on WiFi network. If this roaming is not free, users will be tempted to prioritise roaming on free Hotspots whenever they can. (a) How much money are users ready to pay to use the safe roaming partners of their ISP? Are they aware that free Hotspot may be free for dishonest reasons? (b) A laboratory experiment where people would have to do a trade-off between security and money would be relevant to further investigate this question. This could be achieved through a setup where different test conditions require different fees to pay. An alternative approach could consist in having experiment participants match different usage scenarios with different MNO fees and free hotspots. Indeed, various approaches could be set up here or even combined.

*Use phase: the user uses a dishonest Hotspot 2.0* The network is chosen automatically by the device (a) Are users aware of which policy rule lead them to use this network? Are users aware of the cost of such a use? Are users aware of the modality of this connection (e.g., 3G/4G/WiFi)? Do users trust a connection after having been notified of its occurrence without having asked for it? Do users trust their connection on their MNO's network through a third-party as much as a direct connection? What is the effect of the presence of a seam on the user's trust? (b) These usages are new and the technology supporting them is not widely available

yet, therefore the experiments can not be easily built on existing “usage” standards. Interviews can be performed either in vivo or in a laboratory setup, with people who just experienced some of these situations, to understand what they are aware of in terms of security.

### **HS2.2: The future of Hotspots**

*Selection phase: the user connects to a dishonest Hotspot* (a) Does adding more information about the networks help users to select honest WiFi networks? What is the phishing potential of those new information and services? Are users capable of searching for a network to fulfil a task and end up choosing a service instead? (b) Laboratory experiments where participants would have to choose a wireless network to fulfil a high-stake task are relevant to answer these questions. Networks would expose a range of technical qualities; services would be more or less appealing and related to the task.

### **4.2.5 Conclusion**

This section presents a detailed informal technical security analysis of hotspots. From this analysis, it is possible to identify the various phases of a scenario where the user may affect security. It allows for a better understanding of how each phase may affect the security of subsequent phases or actions.

There are also limitations to our work. The analysis was constrained by the specifications of the documentation that was available at the moment that it was performed. Even though Hotspot 2.0 is considered superior with regard to security, our contribution shows such a system can be attacked and further research is needed. This on the other hand is made difficult by the relative lack of documentation on Hotspot 2.0 at this stage. Moreover, our proposed research questions do not represent a comprehensive list and are rather a selection of questions we consider important to tackle next. There may be other relevant questions to address once we start answering the proposed ones.

## **4.3 Investigating Research Questions**

In the previous Section (see Section 4.2), we studied the human-computer interactions in hypothetical situations where users select one out of several hotspots offering access to Wi-Fi networks. Motivated to discover *where* security can fail, we highlighted the points in the user-interaction protocol where users opt for an open (insecure) network even for tasks that require security, and despite the presence of visual indicators (called *cues*) reminding the insecurity of the choice.

However, to improve the security of those interactions one should rather understand *why* users decide insecurely. Indeed, having a better understanding of the way people take decisions at these *critical action* points may help us to pinpoint the factors that needs to be tackled to lessen the likeliness of users behaving insecurely.

Furthermore, understanding the relative importance of the different factors (e.g., user understanding of the visual cues) that can influence the choice of an Access

Point is interesting because it opens paths for improvements. Identifying factors that are irrelevant to security can help designing remediations to their influence. Furthermore, observing discrepancies between user understanding of the cues and user actions can help designing better cues. Finally, adjusting for irrelevant factors that are considered by users in their choice helps quantifying the real value they give to the encryption between their device and an Access Point. In short, such studies help determining where organization should focus their efforts by providing evidence about whether the priority should be given to the user education, to the enforcement of security policies, to the providing of tunneling tools, or to the improvement of network manager's user interface *et cetera*.

Thus, this section's goal is to investigate this “*why*” question by proving whether or not the factors we hypothesise previously as influencing the user selection and use of Wi-Fi networks. From the different hypotheses we raised in Section 4.2, we choose to investigate two of them:

The first hypothesis is that an attacker can exploit the concept of trust in the Wi-Fi selection phase by claiming legitimacy or by priming users with the word ‘trust’.

The second hypothesis is that the graphical cues that network managers present to the user along with the Wi-Fi networks' names influence the outcome of the users' selection. In particular, we test whether users consider the message carried by the security cues, and if so, we ponder the importance of the user's understanding of the meaning of these cues compared to other factors (e.g., the task users are performing).

### 4.3.1 The influence of trust in the selection of Wi-Fi networks.

In this section, we report on the investigations regarding the influence of words related to trust in Wi-Fi selection. This work presents and discusses the results regarding trust yielded from the experimental setup (survey) described in Section 3.3.

#### 4.3.1.1 Introduction

As already pointed out in the previous section, trust is a catalyst factor in many indirect/remote interactions as the ones daily happening over the Internet (e.g., [75, 31]). Trust, when is well-founded is essential to effective interactions, but when it is ill-founded, i.e. ascribed to an untrustworthy entity, can be very dangerous.

Trust is a concept that have several connotations and that can be easily misunderstood. Therefore, we explicitly choose to adopt Gambetta's definition [74] of trust: ‘Trust is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.’ We are interested in the factors that drive the confidence users have that the network they select will be reliable and behave according to their expectations.

In this context, we consider that trust based solely on the name of a network is ill-founded because a network name is irrelevant when assessing the trustworthiness

Table 4.3: Sociodemographics for the population of the survey for the *trust* condition, and to whole survey

Demographics	Condition <i>trust</i> (n = 59)	Total (n = 99)
Female	36%	<b>45%</b>
Male	64%	<b>55%</b>
Age (average)	27%	<b>26%</b>
High School	19%	<b>22%</b>
Bachelor Degree	49%	<b>49%</b>
Master Degree	20%	<b>15%</b>
PhD	10%	<b>11%</b>
Very comfortable using IT	69%	<b>70%</b>
Somewhat comfortable using IT	27%	<b>26%</b>
Very good IT skills	34%	<b>29%</b>
Good IT skills	37%	<b>46%</b>
Average IT skills	25%	<b>21%</b>

of the network. Therefore, we are interested in understanding whether users use the network name as a trust indicator when choosing a network, and if it is possible for an attacker to influence people’s preferences for his network by nudging them to think about trust.

**Research Question.** (Trust\_RQ): *Does thinking about trust affect participants’ preferences?*

**Experimental setup.** As already explained in Chapter 3, our survey relies on an online questionnaire. From the point of view of a participant assigned to the trust condition, the questionnaire was structured into three parts: (1) the socio-demographics; (2) the ‘general preferences’ part; and the (3) the ‘trust’ part that lists the 12 SSIDs and asks respondents to rate them with **special regard to trust** when connecting/avoiding them (i.e., 1-Not at all trusted, 2-Not very trusted, 3-Neutral, 4-Trusted, 5-Highly trusted). The data collection and statistical analysis are identical to the one used to investigate the influence of the context. That is to say that the collected data were analysed using basic descriptive statistics, followed by specific analysis of variance tests (t-tests [124] and Wilcoxon rank [186] tests) in order to assess the significant differences and between general preferences and the trust condition.

#### 4.3.1.2 Results

As shown in Table 4.3, 59 participants from the on-line survey were assigned to this the investigation of the influence of trust. Figure 4.0 displays general preference and trust results side-by-side for all 12 SSIDs. In general we find a tendency towards higher preference ratings when invoking trust (except for eduroam). This is illustrated by a systematic change in the extremes of the Likert scores, shown in Figure 4.0, change that happens regardless of the name’s properties (existing, open,

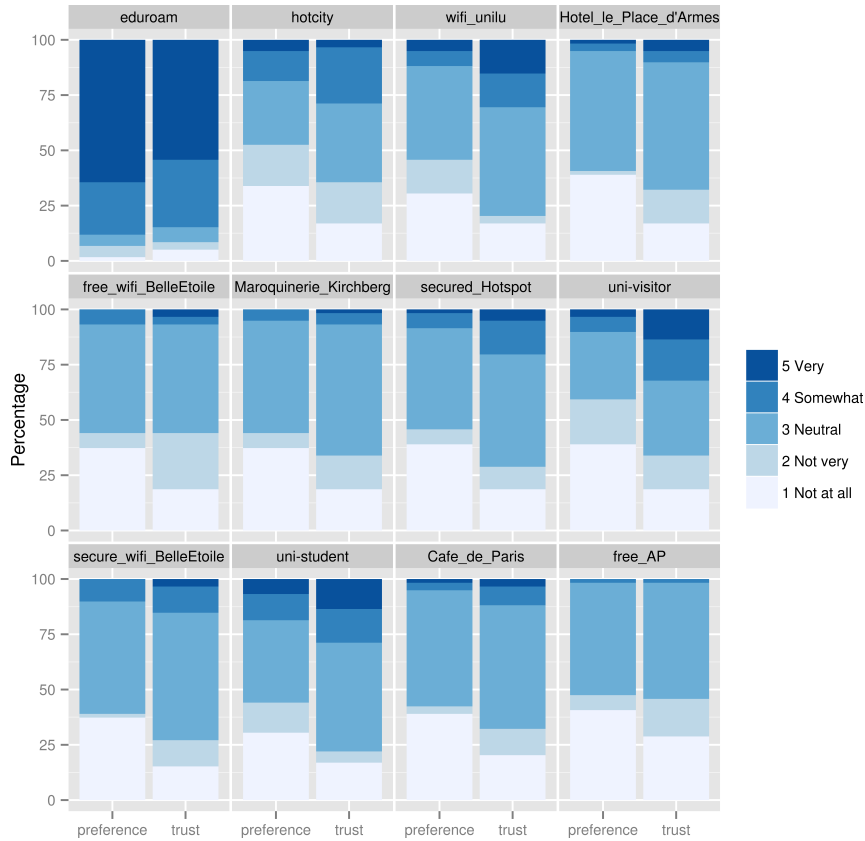


Figure 4.0: General preferences vs. trust. in condition 1 for each SSID.

secure, etc.). A large proportion of the respondents report a neutral preference for each of the wireless network names.

Table 4.4: Statistical significance for the differences between: (a) general preferences and trust; (b) general preferences and trust but for groups G1-G4 (G1-existing; G2-nonexistent; G3-nonexistent and related to security; G4-nonexistent and not related to security).

	Diff. (trust pref.)	Diff. (trust pref. G <sub>x</sub> )			
		G1	G2	G3	G4
Whole sample	0.38**#	0.32***##	0.45*	0.47*	0.44*
Male	0.32*#	0.30***##	-	-	-
≤ 24 years old	0.49**#	0.40***##	0.59*	0.70*	0.53*
> 24 years old	-	0.23*	-	-	-
≤ Bachelor Degree	0.40*#	0.31***##	0.49*	-	0.47*
> Bachelor Degree	-	0.34*##	-	-	-
≤ Good IT skills	0.50**#	0.40***##	0.59*	0.62*	0.58*

(a)

(b)

**Legend:** For all tables superscripts have the following meaning: t-test result: \* $p < 0.05$ ; \*\* $p < 0.01$ ; \*\*\* $p < 0.001$ . Wilcoxon result: # $p < 0.05$ ; ## $p < 0.01$ ; ### $p < 0.001$ .



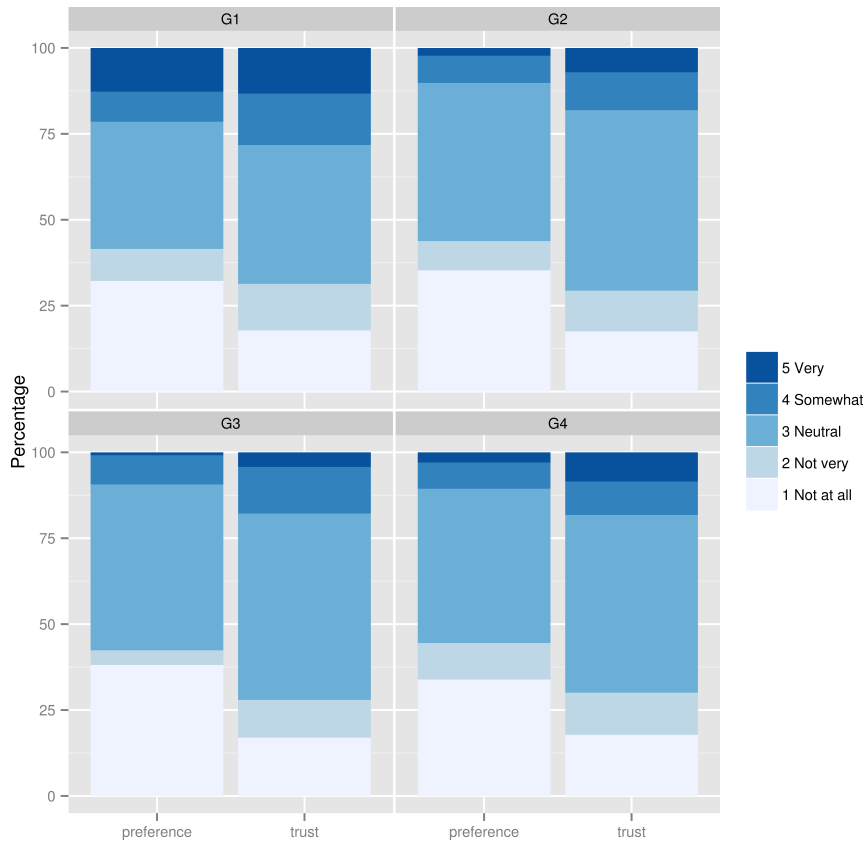


Figure 4.1: General preferences vs. trust for groups G1-G4.

Table 4.4.(a) shows the significant results for the whole sample, indicating that on average the shift from general preferences to trust was towards a more discerning preference (higher positive values).

A similar pattern is shown for the other socio-demographic sub-groups. We also studied more specifically what subgroups of our sample might be particularly affected by this effect. Test results indicate this is true for male participants, for those who are aged 24 years or less, for those who have successfully finished a bachelor degree or less, and for those who consider themselves not very IT literate. Conversely, this means that participants who are not part of these subgroups tend to be more cautious with their ratings in the condition of trust-awareness; our results suggest that age, general education and IT skills contribute to shaping these attitudes.

In addition to the preceding person-centric analysis, we analysed the data more closely under the perspective of wireless network names, allowing us e.g., to better understand whether the formerly described effects apply to all SSIDs or to subsets only. To this end, we grouped wireless network names with regard to our objectives of including them in our study.

Figure 4.1 presents the results between general preferences and trust for the four groups G1-G4 (cf. Table 3.1). Table 4.4.(b) shows the t-test results for the difference in ratings between general preferences and trust, for each of the 4 groups.

The results suggest a strong and systematic effect of trust for G1, for the entire sample, except those participants who describe themselves to be very IT literate.

Table 4.5: Most common reasons related to general preferences (G) and trust (T) for all choices, choices that change to nonexistent names (CPTUN), or to nonexistent names related to security (CPTSN), and that do not change from general preferences to trust.

	All choices (n =53)		CPTUN (n =11)		CPTSN (n =10)		No change (n =18)	
	G	T	G	T	G	T	G	T
Do not use other networks	30	6	4	–	3	–	7	2
Do not know other networks	22	26	2	1	4	1	5	1
Security	13	3	3	1	–	–	2	2
Easy Access	8	–	–	–	–	–	2	–
Trust	3	10	–	3	–	1	–	1

Regarding fake SSIDs (G2), there is still an effect noticeable both for the entire sample and more specifically for subgroups of lower age, lower education and lower IT literacy. This pattern is almost identical for G3 (fake names related to security) and G4 (fake names not related to security). The effects demonstrated for G2, G3 and G4 require further attention as they especially indicate potentially insecure user behaviour. It should be noted that participants who think themselves very IT literate do not demonstrate any effect of trust awareness and it might well be that these participants are aware of trust issues already when considering SSIDs.

Table 4.5 shows the results of the analysis of the open questions. The two most common reasons for participants’ preferences are the fact that they use the networks or they know them, not necessarily because they consider them trusted or secured.

### 4.3.1.3 Security Discussion

Let us look at Figure 4.0. It compares the preferences before and after for the entire sample. Let us focus on the two highest ratings, “very preferred” and “somewhat preferred”: when taken together they indicate a positive preference.

For all network names, with the puzzling exception of “eduroam” (commented in the next paragraph) the preference of a network has increased after people have been asked to think about trust. This seems to indicate that an attacker can gain people’s trust by suggesting trust in the name, at least if he uses names similar to the ones we use in our study. Figure 4.1 shows, in fact, that the increment in preference is almost the same regardless whether the network name exists or not. We therefore conclude that an attacker would be more effective by suggesting or including the word “trust” in the network name itself. If this hypothesis were true, names that hint “trust” should rate better than those suggesting “security” or “freeness”; proving or disproving this claim is left as future work.

We comment now the small drop in trust regarding ‘eduroam’. From the analysis of the open answers it emerges that people said to prefer ‘eduroam’ because they know the network (= have been told to use it); however they said to trust ‘eduroam’ only indirectly (or better comparatively), that is they do not know whether to trust the other networks. Therefore there is reason to believe that people chose ‘eduroam’ by habit, which is a known principle of mental economics. It would be interesting

to test whether people would still use ‘eduroam’ (by habit) in contexts outside the University (i.e., the Shopping Mall), where this network has no reason to exist. This would be an attack to implement with little effort.

### 4.3.2 The influence of graphical cues in the selection of Wi-Fi networks.

In this section, we investigate the role of the graphical cues that network managers usually display along with the network names. There is little research on graphical cues in relation to the security and to the understanding of symbols that network managers rely on. The closest is the research done by Jeske *et al.* who argue that the padlock and signal strength unintentionally nudge people to insecure choices [104]; however they do not explain *why* this happens: are these visual cues unclear and misleading the users? Are they ambiguous and leading users to ignore them? Or are they clear in their messages, but are users choosing insecurely for other reasons extraneous to the cues?

These three questions motivate the work presented in this section. Generally speaking, we could think that users and interfaces are engaged into a sort of *visual conversation* and so it is legitimate to expect it to follow the same principles that rule a constructive and clear conversation. P. Grice, who studied this topic in the philosophy of language, calls them *cooperative conversation* and lists those principles as follows [82]: *quantity* (state what is informative, no more and no less than that); *quality* (don’t state what is false, don’t state what lacks evidence); *relation* (be relevant); *manner* (avoid obscurity, ambiguity, verbosity, and be orderly).

To clarify whether cues are ‘cooperative’ in the sense given above leads to an interesting approach to answer ‘*why* do users choose insecurely?’ in the presence of cues. The approach consists of separating what can be explained as being due to ‘ineffective’ cues from what instead is due to informed choice by the user.

The section describes the particular scenario in which a user chooses a Wi-Fi network. We question whether the common visual cues employed in this task—the padlock and the bars that indicate the signal strength—succeed in communicating their intended message, and we contribute to understanding why.

In the case of Wi-Fi networks, visual cues seem to have been designed with little consideration of the impact they have on user’s behaviour. Indeed, the padlock sign and the signal strength indicators have complex meanings: the padlock sign carries the conflated meaning of encryption and authentication, and the signal strength indicator carries the conflated meaning of reliability and speed of the connection. Therefore, it is not straightforward to understand the impact these visual cues have on the user behaviour and, if the visual cues are ‘cooperative’ for users.

Other authors have studied related questions. As noted earlier, Jeske *et al.* [104] observe that convenience-oriented students *behave* as if the padlock is a barrier to secure choices. They have however not investigated *why* users behave this way. Several key questions thus remain unanswered: does a user *behave* so because they *misunderstand* the padlock or rather because they overlook the padlock due to accompanying factors that force different meanings?

The difference between behaving and understanding is key for us. A user may (a) understand but ignore the cues, and this is after all an *informed* decision. Or they can (b) understand a different message and so take a *misinformed* decision, or they can (c) ignore completely the cue so prefer an *uninformed* decision.

Case (a) suggests that the cue works fine. But (b) suggests that the cue fails and needs a revision, whereas in case (c) the cue is irrelevant, and thus useless. Moreover, in (a) one can still decide insecurely, as well as one can still behave securely in (b) or (c). But, in any of those situations, what nudges the user’s behaviour should not only be searched for in the cue itself, but also in other factors, such as in the presence of other indicators, which influence a cue’s message, or in the task a user is performing, or in the user.

Therefore, this work’s main research questions are the following: Are the padlock and the signal strength and their relative importance responsible for a user’s *informed*, *misinformed*, or *uninformed* decision? Which cues are the most influential in causing that difference, if any? Are the user’s background and different Wi-fi scenarios also affecting the user’s behaviour?



This study builds on observed behaviour of about 1000 participants.




#### 4.3.2.1 Methods and experimental setup

To distinguish the situations where people take *informed*, *misinformed*, and *uninformed* decisions, we need to compare people’s understanding of the Wi-Fi networks’ properties and visual cues relative to the choices they make. Therefore, we conduct a study where we ask participants the following: first, to read the description of a specific scenario setting, a given context and a specific task to perform; second, to choose between different Wi-Fi networks to achieve the task; third, to answer questions about the meaning of the visual cues they encountered; and finally, to answer questions about their knowledge regarding Wi-Fi networks.

What we investigate is whether the choice of a Wi-Fi network depends on the properties of the Wi-Fi network itself and on the specific task to be undertaken. Thus, more precisely, the *dependent variable* we investigate is the participants’ Wi-Fi choice, a dichotomous (i.e., 0/1, wrong/right) variable. As main *independent variables* we choose the presence/absence of the padlock sign (🔒) —supposed to indicate *secure communication*, technically the presence of encryption— and the presence of one of the two signal strength signs (📶 or 📶) —supposed to indicate *quality of connectivity*, technically the strength of the received Wi-Fi signal. These are in fact the properties of Wi-Fi networks typically communicated to the user. In our study we thus display one of the four possible combinations: ‘📶 📶’, ‘📶 🔒’, ‘📶 📶’, or ‘📶 🔒’. In the remainder of this document, for sake of conciseness, we use the terms ‘Encryption’ for *secure communication* and ‘QoS’ for *good connectivity*.

‘Encryption’ (i.e., *secure communication*) and ‘QoS’ (i.e., *good connectivity*) represent also the two meaning dimensions that we assess from our participants in relation to how they understand the cues. We measure how much the participants think a cue means ‘Encryption’ or ‘QoS’, and this is driven by the task a user is involved in; we consider four tasks designed to evoke a need for ‘Encryption’ and ‘QoS’ through context description.

Moreover, to ensure that participants do not avoid networks presented along with a  sign because they do not have a password, we provide a password to half of the sample, aleatorily. This additional independent variable controls the bias that could be induced by the complex meaning that the  sign carries (encryption and authentication), and let us focus on the ‘Encryption’ dimension.

However, one may argue that using the ‘Possession of a Password’ independent variable to this role is limiting and that we could use it to investigate the ‘authentication’ meaning dimension of the  sign. We argue that, this experiment setup is not suitable to do so. Indeed, a treatment where participants are provided a password that works with every network has little to do with authentication and more to do with a ‘magic key’. Therefore, despite the fact that participants are randomly assigned to the different treatments and that we could observe an effect, the effect of possessing a password, drawing conclusions regarding the ‘authentication’ meaning dimension of the  sign from such effect would be incorrect. This is also the main rationale behind the choice of not assessing how the participants understand the  sign along the ‘authentication’ dimension as we could not compare the assessed meaning with the user behaviour.

Besides, we are aware that being provided a password without knowing to which network it corresponds is not a situation that participants can encounter in the real world. However, we argue that, as any treatment, if the ‘Possession of a Password’ treatment leads to a change in the dependent variables, then it is justified to conclude that the change observed is caused by the treatment itself and not by another variable, independently from its real-world potentiality. Therefore, we control for the influence of this variable in the analysis before focusing on the others, and we later avoid to draw real-world conclusions from its observed effects.

Additional *independent variables* that we consider to be important factors to control for are the following: the order of the Wi-Fi network names; speed of appearance over time, i.e., how quickly or slowly the network is listed by the network manager; and the participant’s social and personal background, i.e., tech-savvy *vs* non-tech-savvy users.

To investigate those factors, while maximizing internal validity, we chose an in-between subject study design. Participants were presented only one scenario to avoid security priming of one scenario on the others.

While interviews would be the best option for investigating the understanding users have of the meaning of the visual cues, it is not a good option to investigate the effect of several factors. Indeed, investigating several factors and their relative importance call for the recruitment of an important number of participants that we could reach in-lab. Therefore, the study was conducted online, and extra care has been taken to guarantee the study’s internal validity.

The flow of the study design comprised a socio-demographic questionnaire; the description of a scenario with instructions to select a Wi-Fi network from a given list; several rounds of network selections; an assessment of the understanding participants have for the given cues; and a follow-up questionnaire to assess further attitudes and beliefs about ICT security (e.g., misconceptions and beliefs regarding Wi-Fi networks). In each scenario, we describe for the participant a character they implicitly inhabit and ask him/her what network s/he would select given the context



















1st round	2nd round	3rd and 4th rounds
d1k89 	vputd 	3z6en 
e1hqx 	bra1f 	ko9qb 
auw24 	13zrp 	r6uw4 
2tzza 	37v70 	5crvb 

Figure 4.2: Rounds of choices.

and task to be accomplished. Participants were assigned to respond to 1 scenario out of 4 possible ones; thus the probability of assignment was of .25. Each scenario differed in terms of the requirements the Wi-Fi network should have to complete the task (i.e., combination of ‘Encryption’ and ‘QoS’). Participants had five rounds of choices; each round presented a list of 4 Wi-Fi networks, ordered randomly, each displaying a randomly generated name, a signal strength indicator ( or ) with or without a padlock sign (). Figure 4.2 shows the Wi-Fi networks for the four rounds. To test for consistency we added a fifth round, not shown in the figure: it is one of the presented 4 rounds, randomly chosen.

In the present work, we focus on and describe only the results associated with the third round of network choices. We focus on the graphical cues and their understanding by Wi-Fi networks’ users and leave as future work how the delay, and/or the timing, of the listing of network names affect the Wi-Fi network choices; and also how the sequential order of the Wi-Fi networks makes a difference.

To assess whether users associate the right intended meaning to the cues (‘Encryption’ for the padlock, and ‘QoS’ for the signal strength bar) we ask the participants to express their understanding, using a 4-points Likert scale (Not at all, Partially, Mostly, Completely), of the extent to which they agree that each of the 2 visual cues ( and ) corroborate in meaning with 4 words related to ‘Encryption’ (confidential, protected, encrypted, and private), and 4 related to ‘QoS’ (good signal strength, high-bandwidth, high-speed, and fast).

As mentioned above, we complement the study with additional attitude and belief questions regarding the participants’ use of Wi-Fi networks. For instance we ask such things as their thoughts about whether the padlock sign  means ‘locked out’, and whether they tend to make choices out of convenience. To be clear, our *convenience* variable is a composite of three questions (Cronbach’s  $\alpha = 0.76$ ) and is used as such in our analyses. Additional questions are used to measure ICT skills: these are split into 2 separate variables, *stated ICT skills* (s.ICT) reflecting the participants’ stated ICT skills, and *measured ICT skills* (m.ICT) reflecting how well the participants answered the technical questions. We collected a host of other variables thought to be associated with the Wi-Fi network choice that are not reported on here.

**Choosing the tool for our online survey.** We aimed to have a large number of participants and among a population larger than the one we could reach if we had run our experiment within our University quarters. Therefore, we opted for Amazon Mechanical Turk ([mturk](https://www.mturk.com)), a market place for online work which however offers readily available and substantially large samples of participants. The use of

`mturk` as a tool for social experiments is debated; we are aware of it and of `mturk`'s potential limitations (e.g., [145]) that can harm internal validity. For this reason we took several countermeasures to maximise as much as we could the quality of the collected data. We implemented several quality checks to detect that participants provide answers simply by clicking randomly. Namely, we implemented attention checks, for instance we added choices like: 'I answer randomly and I should not be paid: Yes or No'; we repeated questions several times and we presented them with different wordings; we measured the time participants took to answer each question to detect unusually fast answering which can potentially indicate a low quality of data; we also prevented a participant from participating more than once.

On the positive side, however, `mturk` allows us to recruit participants world-widely, and in the specific case of the US (and we admitted only participants from this country, see later in this paragraph) it is thought to be better representative of the general population than those commonly recruited via university settings [141]. Moreover, evidence suggests that self-reported behaviours gathered with `mturk` are comparable to observed behaviours in laboratory studies [44]. To make our analyses and interpretation of our results easier, we choose to recruit only participants located in the US, where the majority of `mturk` workers do not use the tool as their primary source of income. We ran the study by batch of 100 participants at different times of the day, during workdays and week-ends. Following the guide edited by a community [181] of `mturk` workers, we took great care to guarantee workers' rights of information and privacy, and we paid USD 0.90 for an average of 5 minutes of participation. We collected their age, gender, how comfortable they feel with ICT and their occupation. Occupation categories are organised following the US Bureau of Labor statistic's classification major groups [174]. Optionally, participants can communicate ethnicity related information that follow the US census' interviewing manual guidelines [175].

**The pilot study.** Another issue, not related to `mturk`, but that could potentially challenge the reliability of the data and the internal validity of the study is whether the participants in fact understand correctly what they are presented. In particular, because in theory there is an infinite number of scenarios we could have used to convey and elicit a need for certain Wi-Fi network properties, we had to take special care to pilot test several possible scenarios to identify the ones we ultimately used in our study. For instance, to evoke a task that does not need secure communications or good connectivity, we can ask the participants to picture themselves waiting at a bus stop (no time pressure) searching for a Wi-Fi network to browse the Internet (no need for security), but this scenario could be understood differently by men and women. To guarantee unambiguity in understanding the scenarios, we ran a pilot study using the same tools and settings as the main study that aimed at finding the most intelligible and less biased scenarios. We built 3 different 'vignettes' [71], or candidates, for each scenario, and asked 156 participants to rate how much the task mentioned in the vignettes should comply with several properties. There were 6 properties related to 'secure communications' (confidential, protected, encrypted, secret, masked, and private), and 6 related to the 'good connectivity' (good signal strength, high-bandwidth, high-speed, first-class, responsive, and fast). We analysed

Table 4.6: Chosen vignettes to convey the need for ‘Encryption’ or ‘QoS’ and their limitations.





Scenario	Intended meaning		Displayed text	Limitations
	Encryp.	QoS		
S0-0	0	0	I am sitting in a coffee shop with some friends. As they want to go for dinner later, I use my smartphone to check for a good restaurant. Unfortunately, there is no 3G/4G network available, so I have to use an available Wi-Fi network instead.	QoS is not significantly perceived as needed or not needed, males significantly perceive it as not-needed.
S0-1	0	1	I am a graphic designer intending to show my latest work to some of my friends. Since the 3G/4G connection is failing to retrieve the files, which are rather big, I decide to try an available Wi-Fi network to get some connectivity.	No limitation.
S1-0	1	0	I am waiting at a bus stop and I need to verify whether the check I deposited yesterday has been cleared. I need to use the bank’s application on my smartphone to check the bank account’s balance, but unfortunately there is no 3G/4G. I thus decide to try an available Wi-Fi network to get some connectivity.	QoS significantly tends to be perceived as needed whereas we intend to convey the converse meaning.
S1-1	1	1	I am a government official staying at an hotel. I scheduled an international online meeting. I planned to use the hotel’s Wi-Fi network but the hotels Wi-Fi proved unreliable when I called my family earlier to test the connection. There is no 3G/4G network, so I decide to go somewhere else to find an available Wi-Fi network.	No limitation.





Table 4.7: Sociodemographics profile by scenario.



	S0-0	S0-1	S1-0	S1-1	Total
Gender: Female	41 %	43.4 %	36.1 %	41.2 %	40.4 %
Gender: Male	59 %	56.6 %	63.9 %	58.8 %	59.6 %
Highest ed: High-School	47 %	41 %	42.9 %	40.8 %	42.9 %
Highest ed: Bachelor Degree	41.7 %	48 %	46.4 %	44.6 %	45.2 %
Highest ed: Master Degree	7.9 %	8.2 %	8.7 %	12 %	9.2 %
Comfortable in IT: Not at all	3 %	6.2 %	2.8 %	3.9 %	4 %
Comfortable in IT: Not Very	18.8 %	13.7 %	16.7 %	18.5 %	16.9 %
Comfortable in IT: Somewhat	55.3 %	58.2 %	59.5 %	57.5 %	57.6 %
Comfortable in IT: Very	22.9 %	21.9 %	21 %	20.2 %	21.5 %
Total counts	266	256	252	233	1007

the results of the pilot study with the R statistical software [144] and performed Wilcoxon rank tests [186] to discriminate the vignettes with the best psychometrical discrimination while checking for gender, age, and other social background variable effects. Table 4.6 shows for each scenario: the technical property that it intends to convey (‘Encryption’ or ‘QoS’), the selected vignette, and the limitations we need to be aware of when using it.

In summary, we model the dichotomous outcome (dependent variable) using Logistic Regression [129]: we estimate the conditional probability of choosing the target response option ‘clicking on the network with a  and a ’ net of important independent variables. Our statistical modelling approach is relatively straightforward: firstly, we investigate the effect of the password because we expect it to be an important and significant control; we in fact find evidence of this and thus include it in all subsequent models. Secondly, we investigate the question of whether participants make an informed decision relative to each scenario, and then whether the participants’ answers reflect, in a consistent way, their expressed choice relative to the meaning they attribute to the  and  cues. Finally, we investigate whether the respondents’ choices vary significantly by several basic socio-demographic variables.

### 4.3.2.2 Results

A total of 1090 participants took part in our study. Of these 83 failed the post-hoc data quality and integrity checks, and we remained with 1007 consistent cases. As shown in Table 4.7 and Figure 4.3, our sample is rather balanced with regard to gender. The age distribution has a wide range (56 years). Table 4.8 shows the frequency of clicks (counts) and percentages for the round under investigation in this manuscript, the 3rd. Only 7 participants chose a network with a ; since this gives a too low variability, we excluded those 7 cases and proceeded with our statistical analysis on the 1000 remaining cases that display a .

Varying ‘Encryption’ and ‘QoS’ (independent variables) in order to measure WiFi selection outcomes (dependent variable) may give biased results, because choosing or avoiding network selections marked with a  can occur as an effect of our independent variables or as an effect of simply having a password available or not. In order to control this potential bias, we provided half of the sample with a password. Performing a logistic regression allows to determine if the password is a significant predictor of the outcome ‘clicking on the network with a .

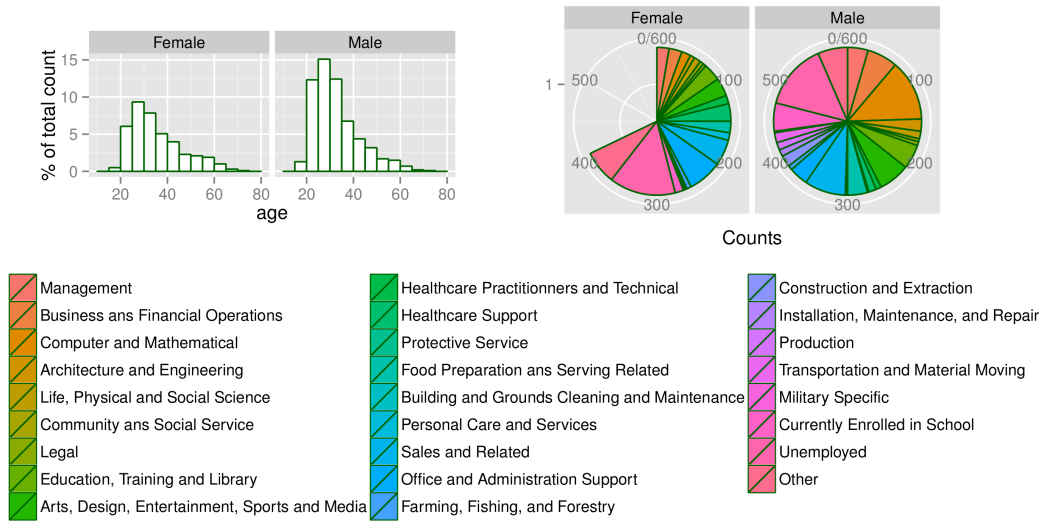


Figure 4.3: Age and Occupation distribution for Males and Females

is an effect based on our independent variables. With a password, odds of clicking on the target are 2.1 times higher (exponentiated coefficient ( $\text{expcoeff}$ )=2.1 with  $p < 0.001$ ). Tested in each scenario, the password effect is significant in S0-0 ( $\text{expcoeff}$ =3.22,  $p < 0.001$ ), and S0-1 ( $\text{expcoeff}$ =4.7,  $p < 0.001$ ).

Table 4.8: Counts and frequencies for the third round of the study.

	counts	frequencies
📶	5	0.5 %
📶🔒	688	68.3 %
📶🔒	2	0.2 %
📶🔒	312	31 %

Table 4.9: Trimmed results of the logistic regression of network selection on password + scenario. (S0-0 reference category)

	Password	S0-1	S1-0	S1-1
p	< 0.001	< 0.01	< 0.01	< 0.001
expcoeff	2.2	2.0	1.9	4.2

As the scenarios evoke the need for ‘Encryption’ and/or for ‘QoS’, we first analyze if the scenario is a predictor of the outcome i.e. ‘clicking on the network with a 🔒’ while adjusting for password. If people understand the meaning of the cues correctly, using S0-0 as intercept: S0-1 (‘QoS’ needed) should not increase the odds of clicking on the target, and S1-0 (‘Encryption’ needed) and S1-1 should increase the odds in the same proportion. The results shown in Table 4.9 prove that scenarios S0-0 and S1-1 increase the odds in the same proportion and that S1-1 nearly increases the odds twice as much. To investigate this result further and to determine if the participants took an ‘informed’ decision, we consider the meaning the respondents associated with their responses. That is to say, we include an interaction term ( $\text{meaning} \times \text{scenario}$ ) and checked the resulting model fit statistics (LR test).

Table 4.10 shows that while the main effects of the meaning dimensions are by large significant, with the exception of the encryption for the 📶 symbol, the LR tests show lack of improvement in model fit by including the interaction terms.

Table 4.10: Exponentiated coefficients of the logistic regressions for the main effect of “Encryption” and “QoS” while controlling for password and scenario. LR tests compare models with and without interaction terms.

Cue	Dimension	Main effect		LR Tests
		expcoeff	p	p
🔒	Encryption	0.823	< 0.01	NS
	QoS	0.727	< 0.001	NS
📶	Encryption	0.860	NS	< 0.05
	QoS	0.826	< 0.01	NS

Table 4.11: Logistic regression results. Tests are performed between the current model and the previous one. AIC is evaluated as well. (\* < .05; \*\* < .01; \*\*\* < .001)

Variables	Step 1	Step 2	Step 3	Step 4	Model fit	
					LR Tests	AIC
Password	2.2 ***	2.4 ***	2.4 ***	2.4 ***	-	1171.88
S0-1	2.0 **	NS	NS	NS	-	1171.88
S1-0	1.9 **	1.8 *	1.8 *	1.8 *	-	1171.88
S1-1	4.2 ***	4.5 ***	4.6 ***	4.7 ***	-	1171.88
Convenience	-	0.10 ***	0.11 ***	0.11 ***	< 0.001	1000.87
m.ICT			Not significant, not added.			
🔒 = Enc.			Not significant, not added.			
🔒 = QoS	-	-	0.91 **	0.91 **	< 0.01	993.25
📶 = Enc.			Not significant, not added.			
📶 = QoS			Not significant, not added.			
age	-	-	-	1.0 **	< 0.01	986.72
gender			Not significant, not added.			
occupation			Not significant, not added.			
s.ICT			Not significant, not added.			
Ethnicity			Not significant, not added.			
n = 1000						986.72

This suggests that the effects of the meaning dimensions do not vary significantly per scenario.

Then we turn our attention to the socio-demographic effects. Age has a significant effect ( $p < .001$ ) as increasing age by 1 multiplies the odds of clicking on the target by  $\text{expcoeff}=1.026$ . Having good measured IT skills multiplies the odds of clicking on the target by  $\text{expcoeff}=1.389$  ( $p < 0.05$ ). Convenience-driven participants are  $\text{expcoeff}=0.104$  ( $p < 0.001$ ) times less likely of clicking on the target. Interactions of convenience with the scenarios are not significant.

Gender, occupation, ethnicity and stated ICT skills don’t have significant effects.

To investigate the predictive power of the independent variables in our model, we conducted a series of logistical regressions in a stepwise fashion. We start with an adjusted model that includes password and scenario, then we add: the convenience, the measured ICT skills, the meaning dimensions, and the socio-demographic variables. Results are presented in Table 4.11 and discussed in the section below.

### 4.3.2.3 Security Discussion

Previous research shows that the  $\mathbb{A}$  can act as a barrier for the user to choose a secure network [104]. This suggests that users are taking a ‘misinformed’ decision, misunderstanding the meaning of that cue. This is actually the case because Table 4.11 shows that when  $\mathbb{A}$  is misunderstood as meaning ‘QoS’, users are less likely to choose the encrypted network.

Our results support that  $\tau$  is the cue that interferes the most with the other cues. That is to say, we were unable to perform any substantive statistical analysis on this particular issue because only 7 participants out of 1007 chose a network with a  $\tau$ : participants avoided the  $\tau$  sign without any regard for the other cues it was associated with or any other contextual factors. We can’t discuss further the weight of its meaning in the decision without statistical evidences, but as participants massively rated  $\tau$  as being the least related to ‘QoS’ we can infer that they took ‘informed’ decisions.

Table 4.11 lists the results of our regression modelling approach and shows the effect of adding other factors one by one. ‘Convenience’ is the *most powerful predictor* of Wi-Fi network selection. We find that being convenience-driven lowers the probability of choosing the encrypted network by 89%. In fact, when we include ‘Convenience’ in our model, it cancels-out the effect of scenario S0-1 (‘QoS’ needed); this effect suggests that the choices made for that scenario are explained by the convenience factor rather than the scenario itself.

‘Scenario’ is the *second most powerful predictor*. For instance, in the final model (Step 4), participants are 4.7 times more likely to choose the encrypted network in S1-1 (‘Encryption’ and ‘QoS’ needed) than in the S0-0 scenario, which is the reference point. But the results also reveal an unexpected behaviour: participants are, almost equally, more likely to choose the encrypted network in both S0-1 (‘QoS’ needed) and S1-0 (‘Encryption’ needed). In S1-0 (‘Encryption’ needed), we can interpret that the participants seek for ‘Encryption’ (still ‘QoS’ can interfere because of the limitations, see Table 4.6), but in S0-1 (‘QoS’ needed) only the need for ‘QoS’ can foster the choice of the encrypted network. Furthermore, still relatively to S0-0, change in odds in S1-1 are more than double than those for S1-0 (‘Encryption’ needed)– this difference suggests that participants confuse ‘QoS’ and ‘Encryption’; and that needing ‘QoS’ contributes to the choice of the encrypted network. Finally, we already observed that the introduction of ‘Convenience’ in Step 2 cancels out the effect of S0-1 (‘QoS’ needed), but this inclusion has a limited effect on S1-1 and S1-0 (‘Encryption’ needed). This suggests that the choice of an encrypted network that is only nudged by the need of ‘QoS’ is fragile; the same choice performed in a scenario needing ‘Encryption’ is stronger. That is to say, even convenience-driven people tend to adopt secure behaviour when the situation calls for it.

We cannot say definitively whether or not the participants’ understanding of the meaning of the cues is the cause of the discrepancies we observe in Step 1’s odds of choosing the secure network for S1-0 (‘Encryption’ needed) and S1-1. As shown in Table 4.11 this is an important factor, but Table 4.10 shows that it does not interact with the scenario and therefore it is not the cause of those discrepancies.

The *third most powerful predictor* is the ‘Possession of a Password’: participants with a password are 2.4 times more likely to choose the encrypted network (see final

step in model). But the effect interacts with the scenario: in a scenario needing ‘Encryption’ participants tend to choose the encrypted network, ignoring whether they have a password or not; but when the scenario does not require ‘Encryption’ it appears that they do not look for an encrypted network, unless we provide them with a password.

The ICT skills that we asked our participants about did not result in significant effects as shown in Table 4.11. Furthermore, we found evidence that knowing what a cue means in terms of the dimensions we asked about, has very little impact on the participant’ decisions. Thus, taking ‘informed’ decisions does not foster a secure behaviour and computer literacy seems to play little role in the decision process. The last significant factor is age, but its effect ends up being nonsignificant.

## 4.4 Conclusion

In the previous sections, we studied the socio-technical security of Wi-Fi Hotspots. We followed STEAL’s operational guidelines, starting by performing a informal technical security analysis to identify *critical action points* that lay in the interactions between a Hotspot and its users, given a specific Threat Model. We then produced potential research questions and imagined experiments to explore the social layers of Wi-Fi Hotspots’ security. Finally, we selected and investigated two research questions: first about user trust, then about the graphical cues displayed by network managers.

We tested how people are biased to choose WiFi access point names when we offer them a pool of names among which there are names of real WiFi networks, names that remind security and trust. Regarding trust, the take away from this study is that networks that use words related to trust may have higher chances of being chosen.

We also explained *why* people choose Wi-Fi networks by investigating how the cues (🔒, ⚡ and 📶) displayed by Wi-Fi network managers affect Wi-Fi network selection. Using a sample of 1000 participants, collected through the Amazon mechanical turk, we analyzed through a series of logistic regressions the relative importance of the various factors associated with the participant’s choice of Wi-Fi network. We shed light on whether users understand and use the padlock and the signal strength visual cues to decide which Wi-Fi network to connect to: they blankly avoid the networks displaying ⚡ because they understand that it is a sign of bad connectivity, but the decision is more subtle when 📶 🔒 and 📶 are competing. The choice of a network displaying a 🔒 is subject to more influences: users who are not convenience-driven tend to pick an encrypted network if they are provided a password or if the task undertaken calls for “QoS”; when needing “Encryption”, all users tend to choose encrypted networks. But our analysis shows that the meaning our participants attribute to the cues and other socio-demographic variables does not explain why our participants choose encrypted networks when the task asks for “QoS”, or even “Encryption”. These results suggest that beliefs and circumstances (i.e., context) are the real motivators behind our participants’ choices, and that even if they take ill-informed decisions regarding the meaning of the cues, they take “informed” decisions with regard to other factors.

In future work, we will seek to confirm our findings reported in this manuscript relative to the other rounds of data collected in our study. We will further investigate how the expressed beliefs of our participants regarding Wi-Fi networks affect their network choices. Moreover, we will investigate more closely the socio-demographic profiles of those who we have identified as being convenience-driven.

Additionally, further study should be conducted to verify the external validity — or real world applicability— of this study. In-lab experiments with users of existing Wi-Fi networks could investigate whether users tend to behave insecurely when facing particular settings. In particular, scenarios where users are given a password to use a particular —trusted— network (e.g., eduroam).

Regarding Wi-Fi hotspot, there is no one-size-fits-all solution. With the implementation of Hotspot 2.0, we recommend that it needs to be better tested for socio-technical security. Although technical security has improved in comparison with the previous hotspot version, many issues still need addressing before its full deployment and usage in parallel with that previous version (which will not quickly disappear). We have provided a series of research questions and experiments to face some of the encountered security problems that industry and research will have to deal with and started to tackle two of these.

Regarding the use of STEAL, we believe that it is important to analyse security of socio-technical systems, but instantiating the whole process takes time and relies heavily on the skills on the analyst. This is particularly true regarding the analysis of social layers. We only investigated 2 factors that could undermine Hotspot's security and we are far from reaching the end of the list of potential user-related issues. Furthermore, STEAL does not provide a way to systematically reuse the discoveries made on one system on another.

We question whether there is another way to study the socio-technical security of a system. A technique that could help us understanding what makes a user fall or not for a STA and devise appropriate defences; but with the additional features being more systematic in the way it links the factors pertaining to the user with adverse effects on a system's security, and that allows to build a body of knowledge on STAs to improve subsequent analyses.

In the next chapter, we lay the foundations of the development of an RCA methodology for computer security. A methodology that could complement or supplement STEAL with a systematic identification of socio-technical vulnerabilities in a system.

## **Part III**

# **S·CREAM, a Root Cause Analysis methodology for socio-technical security**





When I was a kid, they had a saying, ‘to err is human but to really fuck it up takes a computer.’

—Benjamin R. Smith, ‘Atlas’

# 5

## Beyond STEAL: building an RCA for security

### Contents

---

<b>5.1</b>	<b>Introduction</b>	<b>87</b>
<b>5.2</b>	<b>RCA in Safety and Security: Differences and Challenges</b>	<b>89</b>
5.2.1	RCA in Safety	90
5.2.2	From RCA to prediction	91
5.2.3	RCA in Safety and Security: Differences	92
5.2.4	Towards an RCA for Socio-Technical Security: Challenges	98
<b>5.3</b>	<b>Conclusion</b>	<b>100</b>

---

*In this chapter, we discuss the requirements for a Root Cause Analysis for security. We start by introducing RCA techniques, their origin and use in the safety field. Then, we identify the main differences between the safety and security fields that lead to a list of challenges that one should consider when building an RCA for security.*

### 5.1 Introduction

In Chapter 3, we presented a framework—STEAL—that supports the security analysis of socio-technical systems; then in Chapter 4, we used this framework on a use case and identified several caveats. The first caveat is that running STEAL analyses of systems is an expensive process that relies a lot on the analyst’s skills and his capacity to formulate and test relevant hypotheses. Furthermore, STEAL does not provide any way to use the findings made on a system to ease the analysis of another.

The quest for a systematic methodology for the security analysis of socio-technical systems is still open; we believe that we can take inspiration from the safety field because it faced the exact same challenge decades ago, and it has developed methods to tackle those.

Indeed, accounting for the role of a user in a system’s safety incident is a complex matter, and in safety this role is usually studied by applying analysis techniques such as the *RCA* [27]. RCA helps understanding the real reasons for a safety-critical incident/accident occurring in a system operated by humans. Starting from an observed failure that only superficially seems due to human error, RCA guides the analyst to find the reasons for the failure beyond its appearance; it does so by looking for error-inducing technology hosted by the system, poorly designed interfaces and cumbersome interactions with users, and environmental factors that may have affected how users operate the system.

When RCA is used in security, the focus is on the technical details that lead to a security incident; the user’s actions are acknowledged but never investigated. Thus, the analysis of a security failure in systems that interact with humans often ends by pointing out users as culprits and the weakest security links [152].

Such conclusions are short-sighted. Even when exploiting humans as backdoors—we recognise that there are indeed users that execute *security critical actions*, such as clicking on an infected attachment—attackers also rely upon other factors that they can control. For instance, attackers often exploit cognitive biases, use social engineering and persuasion techniques [62], take advantage of ill-designed usability and systems, abuse social norms, or interfere with the social environment by adding noise and exerting psychological pressure. These untangled factors are pre-conditions for the attacker’s ability to trigger what only superficially can be dismissed as ‘human errors’. Understanding the role of these factors and how they can be manipulated by an attacker provides insights on why the security incident occurred in the first place. This is what an informative security analysis should do.

**Motivation.** We want to study security incidents to understand the role of human-related factors in them and we aim to understand how such factors have been (but also, can be) manipulated by an attacker to produce the cascade of events that leads to the success of user-mediated attacks. Besides, we believe that such factors can be understood by using strategies inspired by the RCA practice followed in safety, and that these strategies can help in producing a technique free of STEAL’s limitations. By looking at the ‘human error’ not as a conclusion but as a start and by considering it a symptom calling for an investigation of its ‘root causes’, we think we can retrieve causes that are more informative in their explanations of why and how security fails in the presence of humans and more useful in helping us find ways to enhance security.

The time seems ripe for this kind of investigation. Today, computer security experts acknowledge that systems, users, their environment and their interactions should all be considered together in the security problem [96], and they recognise that users are not always the primary reason for a security failure although they can be involved in it [50]. This statement is supported by research that proves that vulnerabilities may arise when the security design completely misunderstands

the role of users, ignoring basic principles about how people think, reason, and behave/interact with technology [3].

Related works have been done, for instance Cranor *et al.*'s work on security-related communications [42], Curzon *et al.*'s *Cognitive Framework* [45], Carlos *et al.*'s proposal of a taxonomy of human-protocol weaknesses [30], and Kirlappos *et al.*'s findings on users circumventing security policies and controls [113]. All these discuss the role of users in security and, from different perspectives, explain how human aspects may affect security. Our work can be seen as a re-elaboration of these discussions, extended and integrated in our methodology of analysis for the search of the root cause in security, towards a unified mean to address human-related factors in security.

Computer security still misses a strategy of analysis like the RCA in safety where blame is considered the enemy and human error is considered as a symptom, not a cause. This chapter studies what benefits an RCA will bring into the security analysis practices, and what it should achieve. Furthermore, it comments on a list of associated challenges that one needs to tackle when building such a technique for computer security.

## 5.2 RCA in Safety and Security: Differences and Challenges

Before elaborating on the details about RCA in safety and security, it is worth mentioning that the common role of RCA is to provide insights about a complex situation that leads to an adverse event. The primary objective of the RCA method is to identify the root cause of such an undesirable event. This root cause, if removed, prevents the event from occurring. However, sometimes several factors concurrently cause the event, and it would not be justified to point out one of them as the ultimate culprit. Whichever the case, the knowledge gained from an RCA should enable the analyst to write guidelines or rules that implementers can follow to control identified factors and avoid the recurrence of the said event.

To identify these factors and produce controls, an RCA methods usually follows a four-step process (see Fig. 5.1):

**Data collection and investigation** The data collection phase is where an analyst gathers factual information about an incident. The collection and investigation are done objectively and their results should not be dependent on the analyst. The outcome of this phase is a description of the incident.

**Retrospective Analysis** The goal of the retrospective analysis is to determine the causes of the incident. Depending on the RCA method used, this phase can be more or less depend on the analyst's knowledge and experiences. The outcome of this phase is a set of root causes without which the incident could not have occurred.

**Recommendations generation** In this phase, the analyst determines a set of recommendations to prevent the analysed event from recurring. Whatever the

method, this phase is analyst-dependent because he needs to understand the outcome of the analysis and generate sound remediations about what should be done.

**Implementation of recommendations** This is the ultimate phase of a Root Cause Analysis, where the analyst’s recommendations are implemented to prevent additional occurrences of the incident from happening. The outcome is a system freed of its identified caveats.



Figure 5.1: The process usually used for a Root Cause Analysis.

### 5.2.1 RCA in Safety

In safety, RCA has been motivated by the colossal costs in human lives and environmental damages resulting for instance from transportation disasters or nuclear accidents. Such events engendered the development of RCA methods to prevent such horrendous events from recurring, and to make safety-critical systems error-resilient. Even events that are supposedly caused by a ‘human error’ are investigated thoroughly to find out what should be altered or documented in the complex interplay among the human, the system, and the context, in order to prevent the event from happening again.

Indeed, in his accident causation model, Reason [147] shows that ‘*human errors*’ are *active failures* that, when combined with *latent failures*, can transform a simple hazard into an accident. Reason’s model metaphorically describes a system as a Swiss cheese: each slice or layer is a system’s defence against failure and the holes on the slices are the flaws that, when ‘aligned’, create a hazard (Fig. 5.2). A person responsible for an accident is not to be blamed alone, but it is the system as a whole that needs to be investigated because it hosts the fertile ground for the ‘human error’ that caused the accident. Consequently, when searching for the root causes of an accident, one must seek for all the contributing factors and consider ‘human errors’ as manifestations of additional factors that one must also identify.

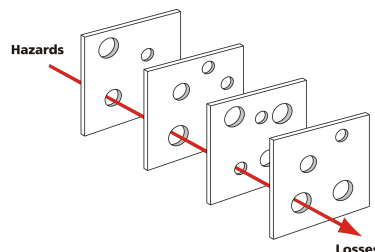


Figure 5.2: Reason’s Swiss cheese model

A common illustration of the contribution of RCA in pointing out design failures that drove humans to provoke disasters is the case of the B-17 Flying fortress, a plane

that was used extensively during WWII. The flying fortresses would sometimes crash because co-pilots retracted the landing gear by mistake while taxiing the plane on the runway. Chapanis [32] found several contributing factors surrounding the taxiing procedure of this aircraft that, in war time, could cause this ‘human error’ to happen, namely:

**Operational** – actions of the aircraft’s controls follow a fixed procedure that could foster mistakes (e.g., before landing: extend the landing gears then lower the wing flaps; after landing: raise the wing flaps [125]);

**Technological** – switches and levers could be confused, since they had all the same shape and were located closely together;

**Environmental** – visibility and light conditions could be suboptimal, since missions could take place at any time of the day, and smoke can be present in the cockpit if the aircraft has been damaged.

**Human** – crew can be stressed, tired or in physical pain, since they were in a war mission.

A root cause analysis of the B-17’s crashes demanded first to identify what factors one can control to avoid accidents from reoccurring. Since nothing could be done to control the environmental conditions, not much to reduce the stress due to the war, and very little to change how to operate the aircraft, the analysis focused on the technological factors. They seemed to be the factors one could control and change. B-17’s switches and levers used to control wing flaps, and landing gears were redesigned to avoid confusion. Nowadays, this new design is enforced by the FAA Regulations that state that the landing gear control must be designed in the shape of a wheel and the wing flaps control in the shape of a wing, and they have to be put further apart from each other [60].

This example enlightens a key aspect of RCA: the findings must allow for actual design of controls that fix the problem. For instance, pointing out that ‘the weather’ is the root cause of an aircraft’s crash is not a suitable answer because there is nothing one can do against bad weather. The root causes of such a crash are to be found in the socio-technical system in which the aircraft operates (i.e., the aircraft, the crew, the people on the ground, *et cetera*) that should be designed to cope with bad weather conditions.

### 5.2.2 From RCA to prediction

While RCA is used to understand how an event has developed, the safety field also make use of techniques to predict the performance of a system that contains a human component. So instead of working from the observed consequences to the root causes, the prediction aims at showing how a scenario—a system failure—may develop depending on human performances, and at computing the probability of recovering from this failure.

Predicting how an event can unfold is highly dependent on the description of the context, tasks, and failure modes. Potential paths that an actual event can

follow are usually represented in binary trees called event trees (see THERP for instance [167]), where branches represent what happens when an event (a leaf) is a success or a failure. Eventually, probabilities are computed for each outcome and recommendations are produced to enhance the reliability of the system.

This prospective approach is used in Human Reliability Assessment. It relies ‘heavily on process expertise to identify problems and HRA methods themselves or expert estimation for quantification’ [21]. The overall process for a prospective analysis follows the same process introduced earlier for an RCA (see Fig. 5.1), except for the analysis.

### 5.2.3 RCA in Safety and Security: Differences

The RCA methods in safety that helped engineers create almost defect-free operations of planes and nuclear plants have come at the price of analyses of past events. In support of these activities, regulations and laws have been set to enforce the use of good practices and protocols.

Security seems running behind in this matter. It is only recently that the conscience for the life-threatening dimension of security incidents and other dramatic consequences they may have seems to have arisen in the security field. Data collection, incident response, and forensics are, at the time of writing (February 2016) slowly getting imposed by law [39], but these practices are still mostly embraced on a voluntary basis.

There are a few key differences between an RCA approach in safety and an RCA approach in security that emerges preponderantly. Such differences (see the next sections) make migrating RCA from safety to security be not a straightforward task; they indeed bring up several challenges that need to be addressed and resolved before we can set a framework of analysis. We have identified a non-exhaustive list of eleven differences that we think are the most relevant for our approach. They appear at different steps of the RCA process introduced in Section 5.2:

#### 5.2.3.1 Data collection and investigation

(D<sub>1</sub>) *In safety, there is an established process to collect structured evidence to be used for root cause analysis. In security, this process is not well-established and data are often unstructured.*

To track back the root cause of an event, one needs to get the relevant data. In safety, for example in aeronautics, to achieve the collection of these data, aircrafts are monitored in-flight by the *Air Traffic Control* through the measurement (e.g., radar) and communication (e.g., ADB-S, ACARS) of different parameters. Additional data (e.g., voices and screen contents) are recorded into ‘black boxes’ (i.e., *Cockpit Voice Recorder* and *Flight Data Recorder*) to enable the forensics after an accident. When an event occurs, investigations are supported by the use of taxonomies (e.g., the Accident/Incident Data Reporting taxonomy (ADREP-2000)[98]) where sub-events, and observed human behaviours are coded.

In security, corporate networks have the same kinds of data collection and protection in place: *Intrusion Detection Systems* and *Anti Viruses* look for *Indices of*

*Compromise*—or, ‘any piece of information that can be used to search for or identify potentially compromised systems’ [57]—to trigger an *Incident Response* when a live attack is detected, and logs are collected for a later forensic investigation if an intrusion is detected after it occurs. But, while the data that constitute *Indicators of Compromise* are well-defined for sheer technical attacks (e.g., IP addresses reached, signature of binaries, registry keys changed, *et cetera* see OpenIOC [127]), information about STAs are less structured and often described in text. Some taxonomies that consider the human exist though, the Vocabulary for Event Recording and Incident Sharing (VERIS)[177] taxonomy is used to structure information about security incidents. It contains, among numerous others, fields about human actions (and errors), demographics, social engineering techniques used, and root causes of events. These kinds of taxonomies do not intend to describe observed human behaviour and environment in detail to support an analysis but to describe the result of an expert’s analysis.

(D<sub>2</sub>) *In safety, there are no malicious actors and incidents happen because of general malfunctioning. In security, incidents are caused by attackers whose skills and capabilities may be subtle and even unknown.*

In safety, there is no such role as an attacker: safety-critical accidents are not triggered maliciously and are usually coincidental events<sup>1</sup>. In contrast, a security incident (i.e., an attack) comes from the attacker’s intention to benefit from the system. Concepts like *Attacker’s goal*, *Asset*, *Threat*, and *Vulnerabilities* are absent from the safety vocabulary and some destinations of the data collected in security are completely alien to the safety field. Such destination is Threat Intelligence [105] that consists of organisations sharing data about current and past attacks to enhance their situational awareness (e.g., Malware Information Sharing Platform [130] or the VERIS Community Database [176]). Therefore, in addition to supporting the analysis of past events, data collection destined to an RCA for security should support Threat Intelligence, and foster the dissemination of concrete observable artifacts of user-mediated attacks. These sophisticated indicators would enable the surveillance of networks, systems and users for known patterns of attacks that exploit the human component of a system.

(D<sub>3</sub>) *In safety, accidents to be investigated usually take place in well-known and well-defined settings. In security, we face much more heterogeneous contexts.*

In safety, there is usually an operator performing a task (e.g., manipulating a lever) that is part of a procedure (e.g., checklist), in a fixed technological environment (e.g., hardware and Human–Computer Interaction), and a known operational context (e.g., time of the day). When investigating an accident, an analyst knows

---

<sup>1</sup>This tends to change though, as explained in Chapter 2, there is a growing interdependency between both domains with security growing in safety requirement and, maybe also vice-versa. Safety gradually becomes more and more dependent to the security of IT infrastructures, to the point that security becomes a requirement for safety. Indeed, as stated by Kuntze *et al.* in [118]: ‘Safety incidents are accidental, not malicious, in nature. We agree that security incidents are mainly malicious. Therefore, because security incidents are now a probable cause of safety incidents, safety incidents can no longer only be classified as accidents’.

what the operator was doing at the moment of the accident, what was the ‘prescribed behaviour’ (i.e., what behaviour was expected from the operator) and what was the expected ‘performance’ for this task. Finding the root cause of an accident can then often be reduced to sort out the factors that caused a ‘deviation from the prescribed behaviour’ or a ‘degradation in the performance’, and the remedy can be implemented by redesigning the human-machine interface or revising the procedure.

In security, depending on the attack under scrutiny, this information may not be available to support the investigations. Indeed, an RCA for security will face a much more heterogeneous set of contexts, ranging from investigations on attacks on industrial systems where information is available, to investigations on web-applications with little data to support the analysis. Furthermore, where in safety investigations can encompass interviews and on-site collection of evidences, the security field is not used to perform this kind of investigation for security incidents. Most data are machine generated and almost no contact occurs with the end-user (although this tends to change, see for instance Slack’s anomaly detection system [93]).

### 5.2.3.2 Analysis

*(D<sub>4</sub>) In safety, RCA techniques are widely used and the human component is a central part of practices. In security, the use of RCA methods is often advocated but lacks human-related insights.*

Information Security Management frameworks such as COBIT [99] and ITIL [11], like other information security best practices, already advocate the use of RCA to find out what went wrong with a system after an incident. Techniques such as the 5 Whys [122] used in combination with Ishikawa diagrams [100] or Event and Causal Factors Analysis diagrams are advised, but they have several shortcomings. The main such shortcomings being the lack of objectivity and diversity of the analyses. Indeed, as these techniques offer little guidance, their results are limited to the analyst’s knowledge and are different from analyst to analyst. These techniques lack the support to help the analysts to explore causes that they could not think about. In this regard, Verizon’s Data Breach Investigation Report [26] that advises to perform 5 Whys analyses, is careful when it comes to mentioning root causes and prefers the term of ‘victim’s critical omissions’ because ‘even with a detailed technical report, the actual root cause typically boils down to process and human decision making’. Thus, the RCA methods used in security acknowledge human actions in their investigations but fail to examine the causes and factors driving these actions.

*(D<sub>5</sub>) In safety, a ‘human error’ is a well defined concept that can be the starting point of an analysis. In security, ‘human error’ is considered a system’s failure mode that does not call for investigations. Considering every user’s negative impacts on the system’s security—including those resulting from an attack—as ‘Human Error’ to use safety methods is debatable.*

One can argue that safety’s ‘human error’ and the techniques used to understand it are irrelevant in security because the behaviours that caused an attack to succeed



are not ‘human errors’, but the result of the attack or a deception. We choose to adopt Strauch’s definition of ‘Human Error’[163]; a ‘human error’ is ‘*an action or decision that results in one or more unintended negative outcomes*’. Thus, in a security analysis setting, these negative outcomes are the loss of some security properties. As we will see later, we consider the loss of security through user exploitations as system failures; moreover, we consider that we can use safety-inspired techniques to tackle such failures. In this setting, the attacker is no different from the wind for an aircraft’s pilot: an additional element that the user has to cope with and that can lead the user to err, and commit actions that lead to adverse events. Thus, safety’s retrospective analysis methods can be used to identify the contributors to ‘human errors’ in security.

The conclusion is not the same for a prospective analysis though. Security practitioners are used to consider the attacker’s point of view to find likely attacks. Indeed, they often find possible attacks by posing hypotheses about attacks, and potential factors that could be manipulated to build a path towards a goal. This usually leads to the construction of Proof of Concepts—or in STEAL, experiments—in which potential attacks, and the exploitation of identified factors are demonstrated. As explained in Section 5.2.2, in safety, these ‘what if’ scenarios are analysed through prospective analyses to predict the consequences of a random event, and the probabilities of different outcomes depending on human performance. But in security, there is no gain in knowing how an event will unfold or what is the probability of an outcome, because an attack’s success is a terminal failure from a security point of view, and it can not be recovered, no matter how the user performs afterwards. Conversely, identifying socio-technical vulnerabilities in a system—what are the entry points to an adverse outcome—is a success. This difference in the consideration of ‘Human Errors’ makes prospective analysis for security, or the study of the forward propagation of a ‘human error’ in the system straightforward. Indeed, we are not interested in predicting the probability of occurrence of an adverse event; predicting its potentiality along with its contributors and pre-conditions is enough. The main rationale behind this is that if a ‘human error’ aligns with an attacker’s goal, there is a potential threat that needs to be contained.

Furthermore, where in security, the distance to the attacker’s goal constitutes an heuristic to traverse a tree of possible consequences of an event, in safety, the duty of cutting branches of the tree of possibilities is left to the expertise of the analyst. Consequently, without proper heuristics or an expert guiding the analysis, safety’s prospective analyses lead to combinatorial explosions of outcomes for an event.

This differences between RCA in safety and in security hint that retrospective analysis can easily ported from safety to security, whereas prospective analysis may need a different approach altogether.

(D<sub>6</sub>) *In safety, there is always some root cause that can be isolated for an incident.  
In security, the root cause of the success of an attack is always the attacker.*

If we keep in mind that a root cause is the one cause that, if removed, prevents an incident from happening, then for a security incident the root cause is always the attacker. Thus, an RCA for security does not search for a root cause, but for contributors—or factors that the attacker could have manipulated to produce the

adverse event. This property of the RCA contributes to avoiding the stereotypical blaming putting all contributors (e.g., environmental, technological, psychological, *et cetera*) on the same level of importance. We expect this choice to broaden the scope of the search for contributors and foster the analyst's objectivity. Indeed, an RCA methodology that operates this way could prevent the confirmation bias that is a threat for any analyst-based analysis (see for instance Heuer's work on psychological factors' influence in intelligence analysis [89]).

(D<sub>7</sub>) *In safety, an analysis begins from the terminal point of failure: the observable incident. In security, an attack/incident can be an intermediate step leading to other attacks/incidents. Therefore we might not be able to observe the factual consequences of an attack/incident on a system.*

In safety, an incident can be combined or followed by other incidents to create the final incident that calls for an investigation by RCA. This last incident of the chain is the final state of the system, the system failure state. In security, an incident can actually be an attack's step and open paths to further attacks, including the user or not. Indeed, the effect obtained on a system by the attacker can provide him with new capabilities that could help him to further gain privileges or harm the system's security. While this does not imply anything on the RCA per se, it has implications on the results of the RCA: the results need to enable computer security techniques to investigate the implications of these new capabilities.

### 5.2.3.3 Recommendations generation

(D<sub>8</sub>) *In safety, removing the root cause prevents the incident from reoccurring. In security, where the root cause is the attacker (see (D<sub>6</sub>)), technical controls can be applied on the attacker to reduce its capabilities, and socio-technical controls can be applied on the other contributors.*

Where RCA in safety follows Reason's Swiss cheese metaphor (see Section 5.2.1), computer security utilises a derived notion of 'defence in depth' strategy [135], where a threat is contained by several layers of barriers. An attacker who successfully performs an attack on a system breaks through all these layers, and when this attack is a socio-technical attack, it breaks through social layers. As introduced earlier, human-related factors in safety were extensively studied for decades [32], and methods have been created to use the knowledge gained over the years [167]. In security, human-related factors are only studied for roughly 15 years [184], with little progresses [150], and no systematic methods to solve human-related issues are available. Thus, in the case of a STA, an attacker traverses social layers that are well-studied by the safety field, and technical ones that are well-studied in computer security.

Considering that an attacker performs a STA by manipulating the contributors to an adverse event, we are interested in producing recommendations to control these contributors and prevent the attack from being repeated. But as explained earlier, producing recommendations for computer security is a complex matter, because the attacker strikes on several layers and, depending on the layer, the means to

thwart the attack is completely different. For instance, on the technical side, formal methods can be used to prove that, given a Threat Model, certain security properties are met on a system; while education can be delivered to the end users to cover the social side. We argue that the added value of an RCA for socio-technical security is to complement existing computer security methods. Thus, we are not interested in finding contributors and producing recommendations related to the technical layers, but in the social layer, and its interactions with the technical ones. Thus, where in safety, an RCA is used to tackle social, technical, and organisational issues, an RCA for security must focus on the blind spot that computer sciences methods have for the human component, and propagate findings to the methods used for the technical layers. By providing a Threat Model closer to reality for instance.

(D<sub>9</sub>) *In safety, an adverse event, being coincidental, may never reoccur on similar systems. In security, an attack incident will reoccur because attackers actively probe similar systems to recreate it. The sharing of recommendations is thus critical.*

In safety, recommendations to an incident can be shared or enforced by regulations to prevent the incident to coincidentally occur on similar systems. In security, the risk of re-occurrence is different because it is a malicious actor who provokes the incident, and this actor actively probes other systems for the existence of the incident's contributors in order to exploit them. Therefore, when an attack is observed on a system and recommendations produced, others systems should also be considered, and the recommendations should be designed to be shared.

#### 5.2.3.4 Implementation of recommendations

(D<sub>10</sub>) *In safety, people involved in incidents that require an RCA are mostly trained professionals (e.g., pilots, air traffic controllers, power plant operator). In security, the corresponding people are much more diverse with regard to their relevant skills and knowledge (e.g., children, bank employees, elderly people, medical doctor). Furthermore they can have motives and concerns unrelated to security.*

Where in safety, some recommendations can be implemented in regulations, laws and organisational policies, implementation of recommendations is less straightforward in computer security. Standards do exist for companies. For instance, companies need to comply with PCI-DSS [143] to process credit card information. But as security is now integral in every business, segmented standards do not appear to be a suitable solution. Umbrella laws exist to push companies that handle personal data to be liable for data breaches (see, for instance, in Europe [39]), and 'to implement security measures that guarantee a level of security appropriate to the risk presented' [137], but these texts do not provide the level of details found in aeronautics or nuclear safety regulations.

Individuals can be professionals, and professionals follow training and have to follow the rules edited by their hierarchy. Regarding other individuals, they may have to follow a compulsory training to get a licence for an activity, for instance,

driving. When an activity is not subject to any licence, information can be provided to the end-user to foster a safe behaviour, or the environment is altered to add safeguards to lower the consequences of an adverse event (and this is enforced by law). For instance, stickers are placed on the walls inside a building to inform people of the safest route to get out of it, and seatbelts are placed in cars to avoid being ejected in case of accident. In security, recommendations on the environment are similar but controlling user behaviours has to follow other routes than enforced discipline, because even in professional settings, coercion does not work well [113]. Therefore, to control human-related contributors, we can resort to education [138], persuasion techniques [87], behaviour change theories (for instance, see Barrier Analysis [48]), nudges [180], or even gamification [49].

( $D_{11}$ ) *In safety, root causes are identified and controlled. In security, it may be impossible to control all identified contributors that will be actively manipulated by the attacker.*

In safety, a root cause can usually be identified and reliably controlled. But, in security, the attacker can find its way around a control, and expose the corresponding contributor. Therefore, complementary computer security methods should be used to account for the consequences of an attack on a contributor, even if the contributor is controlled. Furthermore, attacks performed on a system can be replayed on similar—but not identical—systems, and variations of attacks can emerge, using part of the same set of contributors. Thus, in security, it is important to consider the implementation of recommendations as one barrier for an attack, not a remedy.

## 5.2.4 Towards an RCA for Socio-Technical Security: Challenges

The differences we have highlighted for each step of the RCA in Section 5.2.3 suggest that we cannot easily migrate RCA from safety to socio-technical security unless we solve a few challenges.

### 5.2.4.1 Data collection and investigation

As in safety, before analysing an incident, data should be collected. However since there is no established process to do so ( $D_1$ ), and we deal with an attacker and aim at fostering Threat Intelligence ( $D_2$ ), and since the context is unknown ( $D_3$ ), the first challenge we have to face is the following:

( $C_1$ ) ***Addressing the lack of knowledge and structured data:*** *The challenge is to compile and format the needed factual information about the investigated attack to allow for the RCA to be performed. In particular, it is important to describe what the attacker does and what are the attack's effects on the user and on the system's security. Furthermore, The RCA should provide precise information regarding the data to collect and the Indicators of Compromise of a STA.*

#### 5.2.4.2 Analysis

As in safety, an RCA for computer security will have an Analysis phase. As previously stated, this phase should encompass the human's erroneous action ( $D_4$ ) and ( $D_5$ ), provide contributors more than a sole Root Cause ( $D_6$ ), and support the analysis of sub-sequent attacker's moves ( $D_7$ ). Furthermore, objectivity and systematicity of the analysis are part of the motivations of this work; therefore, the second challenge is:

- ( $C_2$ ) **Investigating Attacks:** *The RCA for security must output a set of contributors and human-related factors that are likely to explain the success of attacks, or potential attacks. Additionally, the analysis should safeguard against one inherent shortcoming of RCA: the possible lack of objectivity.*

#### 5.2.4.3 Recommendations generation

Providing recommendations as they are produced in safety does not pose any challenge and is achievable by an RCA for security. However, as explained in Section 5.2.3.3, providing recommendations is not enough to thwart Socio-Technical Attacks. Indeed, in addition to recommendations, an RCA for socio-technical security should support existing security methods ( $D_8$ ) and the sharing of information ( $D_9$ ). Therefore, the third challenge is:

- ( $C_3$ ) **Creating reusable knowledge:** *To integrate with existent computer security's techniques, the RCA technique should provide direct links between the attacker's capabilities and their effects on a system's security. The end goal is to be able to augment a said threat model with capabilities that an attacker can gain by performing user-mediated attacks allowed by the threat model.*

#### 5.2.4.4 Implementation of recommendations

Implementation of recommendations as provided in safety is also achievable by an RCA for security. But, as explained by ( $D_{10}$ ), the controls act upon contributors and not the root cause, and ( $D_{11}$ ) that we may not be able to control all contributors, there is a need of synergy between social and technical methods to thwart further attacks. Therefore the last challenge is as follows:

- ( $C_4$ ) **Match patterns of known attacks:** *The RCA, in addition to the retrospective analysis of past attacks needs to provide a socio-technical security analysis where, from a system's description, socio-technical vulnerabilities, along with their contributors are listed.*

#### 5.2.4.5 Subsidiary challenges

A subsidiary challenge emerges from the dynamic dimension of security:

- ( $C_5$ ) **Being flexible:** *As computer systems, the threats that they are exposed to, their practices and the research surrounding these topics evolve continuously, the method should be flexible enough to adapt to new threat, attacks, and technologies.*

## **5.3 Conclusion**

In this chapter, we introduced RCA techniques used in the safety field, and explained how inspiration from these techniques to perform socio-technical security analysis could be beneficial for complementing the framework introduced in Chapter 3.2: STEAL. Several differences were identified between safety and security to produce a list of five challenges that one needs to tackle in order to build a useful RCA methodology for security. In the next chapter, we present our methodology—S·CREAM—that aims at tackling all these challenges and providing computer security with a systematic tool to improve effective security of socio-technical systems.

*Eliminate all other factors, and the one which remains must be the truth.*

—Sir Arthur Ignatius Conan Doyle, ‘The Sign of the Four’

# 6

## S·CREAM: an RCA for socio-technical security

### Contents

---

<b>6.1</b>	<b>Introduction</b>	<b>102</b>
<b>6.2</b>	<b>S·CREAM’s process and concepts</b>	<b>103</b>
6.2.1	Overview	103
6.2.2	The failure of the generic RCA process	103
6.2.3	S·CREAM’s overall process	104
6.2.4	First step: Data collection, and investigations	105
6.2.5	Second step: Retrospective Analysis	107
6.2.6	Third step: Generalisation	108
6.2.7	Fourth step: Security Analysis	110
6.2.8	Controlling socio-technical vulnerabilities	112
<b>6.3</b>	<b>S·CREAM’s implementation</b>	<b>112</b>
6.3.1	Implementation Choices	113
6.3.2	Implementing the <i>Data Collection, and Investigations</i> step of S·CREAM	114
6.3.3	Implementing the <i>Retrospective Analysis</i> of S·CREAM	115
6.3.4	Implementing the <i>Generalisation</i> step of S·CREAM	116
6.3.5	Bootstrapping S·CREAM’s catalogue of Attack Modes	117
6.3.6	Implementing the <i>Security Analysis</i> of S·CREAM	121
<b>6.4</b>	<b>S·CREAM’s companion tool: <i>S·CREAM assistant</i></b>	<b>122</b>
6.4.1	Main functions	122

6.4.2	Additional features . . . . .	126
6.4.3	Technical implementation . . . . .	126
<b>6.5</b>	<b>Conclusion . . . . .</b>	<b>127</b>

---

*In this chapter we present S-CREAM, an RCA technique for security. First, we introduce the technique’s process and concepts, then we detail the implementation of S-CREAM along with a companion tool: S-CREAM assistant. Part of the content of this chapter appears in ‘In Cyber-Space No One Can Hear You S-CREAM - A Root Cause Analysis for Socio-Technical Security’ by Ana Ferreira, Jean-Louis Huynen, Vincent Koenig, and Gabriele Lenzini published in the Proceedings of the 11th International Workshop on Security and Trust Management (STM 2015, Vienna, Austria) [67].*

## 6.1 Introduction

In Chapter 5, we introduced RCA techniques in safety and the challenges that an RCA should meet to be beneficial for the field of security.

In the present chapter, we propose our own RCA for security that is inspired by one particular RCA technique: the Cognitive Reliability and Error Analysis Method (CREAM) [91]. The resulting methodology, called S-CREAM (a contraction of Security and CREAM), is a complete revision of the technique that we extended and adapted to work in security.

**Contribution.** We devise a method, S-CREAM, which when applied to a specific security incident, helps analysts investigate how attackers may have pushed users to perform hazardous actions causing the incident. This capability to identify the factors that would have potentially contributed to the success of an attack is S-CREAM’s main contribution and is inspired by the RCA techniques found in safety. However, to integrate S-CREAM into the landscape of computer security tools and methods, we augment it with subsidiary capabilities. S-CREAM can reuse knowledge on attacks, thus allowing for the identification of weaknesses in other systems before such incident recurs. This new capability provides the two following modalities, depending on how analysts apply our method: (a) applied on a set of analysed attacks, S-CREAM helps compile a catalogue of vulnerabilities, called *Socio-Technical Attack Modes (AMs)*, which attackers can exploit to manipulate user’s actions; (b) applied over a specific system, S-CREAM helps investigate the system’s resilience against a specific threat model.

**Outline.** This chapter covers our implementation of an RCA method for socio-technical security —S-CREAM, and its technical implementation in a tool — *S-CREAM assistant*. The differences introduced in Section 5.2.4 lead to the challenges that S-CREAM must consider and address. Section 6.2 introduces a high level description of S-CREAM’s concepts and process. In Section 6.3, we explain



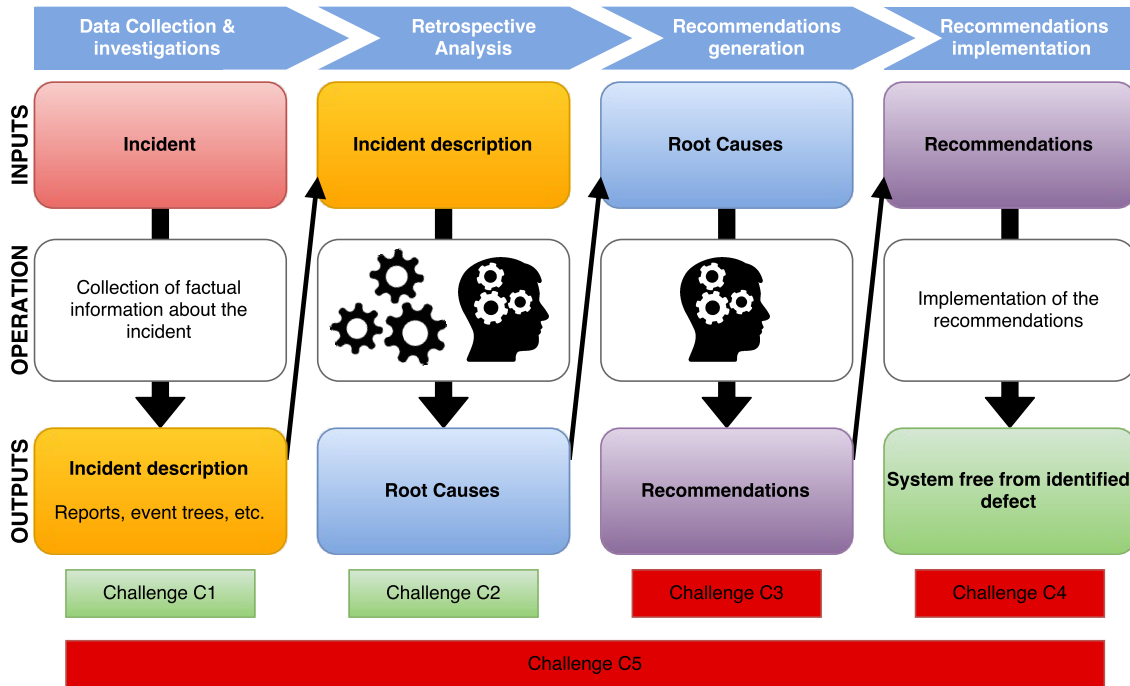


Figure 6.1: The generic process of an RCA against the challenges introduced in Section 5.2.4.

the method's internal mechanisms in detail. Eventually, we present *S-CREAM assistant* in Section 6.4.

## 6.2 S-CREAM's process and concepts

### 6.2.1 Overview

Building an RCA for a socio-technical security analysis means addressing the five challenges listed in Chapter 5. We show that considering these five challenges leads to a new process, which we obtained by modifying the generic RCA process. Figure 6.2 sketches the reviewed process, whereas Figure 6.1 recalls the original RCA process. In both pictures, we show how the process addresses (or fails to address) the five challenges.

The steps structure S-CREAM's high-level work flow, successively detailed in Section 6.3. The reviewed process is still made of four steps, each addressing one challenge, while the fifth challenge is addressed by the all steps taken together.

In the next subsections, we comment on why addressing the challenges introduced in Chapter 5 requires S-CREAM to follow a non-standard process. We also comment on a high-level description of S-CREAM.

### 6.2.2 The failure of the generic RCA process

Figure 6.1 shows how the generic process of an RCA would fail, if used for security to address challenges  $C_3$ ,  $C_4$ , and  $C_5$  (coloured in red in the Figure).

The first two steps of the process seem to be, in appearance, easily adaptable to security analysis. The first step could collect data about an attack and its indicators as stated in  $C_1$ . The second step could implement a customised retrospective analysis that identifies contributors to STAs, and address  $C_2$ . However, the process is unable to address  $C_3$  and  $C_4$ . Used without adaptation, the third step would take the root causes yielded from the retrospective analysis of an attack and generate recommendations for this attack. Eventually, these recommendations would be implemented to prevent the occurrence of this, and only this, precise attack on a system. However, such a generic RCA step is not informative enough to address  $C_3$  and  $C_4$  because it is too specific to the system under investigation. The generic process of an RCA is not designed to produce recommendations that can be transferred to other systems to impede the presence of the socio-technical vulnerabilities enabling the attack.

### 6.2.3 S-CREAM's overall process

The process is designed to be followed by a security analyst, a person investigating attacks or a system's socio-technical security.

As for the generic process, steps are backward-dependent. Each step builds on its preceding step, and thereby, depends on the successful completion of the preceding step. For instance, one cannot obtain a list of the possible contributors to an attack without first describing the attack. This dependency is depicted in Figure 6.2 and read as follows: each step requires all its neighbour steps on the left to be executed beforehand.

For each step, we describe its *concepts* (what it is about), *input* (what it needs to operate), *output* (what it produces), and *operation* (how it operates). We supplement the exposition by referring to the following toy example of an attack wherever necessary:

The setting is a system  $S_1$  on which a user can perform an action  $A_1$  following a request for the identity  $I_1$ . An attacker aims at executing an action  $A_1$  on a system  $S_1$ . To do so, it sends a message that imitates the graphical identity of  $I_1$ , and that contains the request to perform the action  $A_1$  on  $S_1$ , to a user. This user confuses the attacker for  $I_1$  and performs the action  $A_1$  on  $S_1$  on behalf of the attacker.

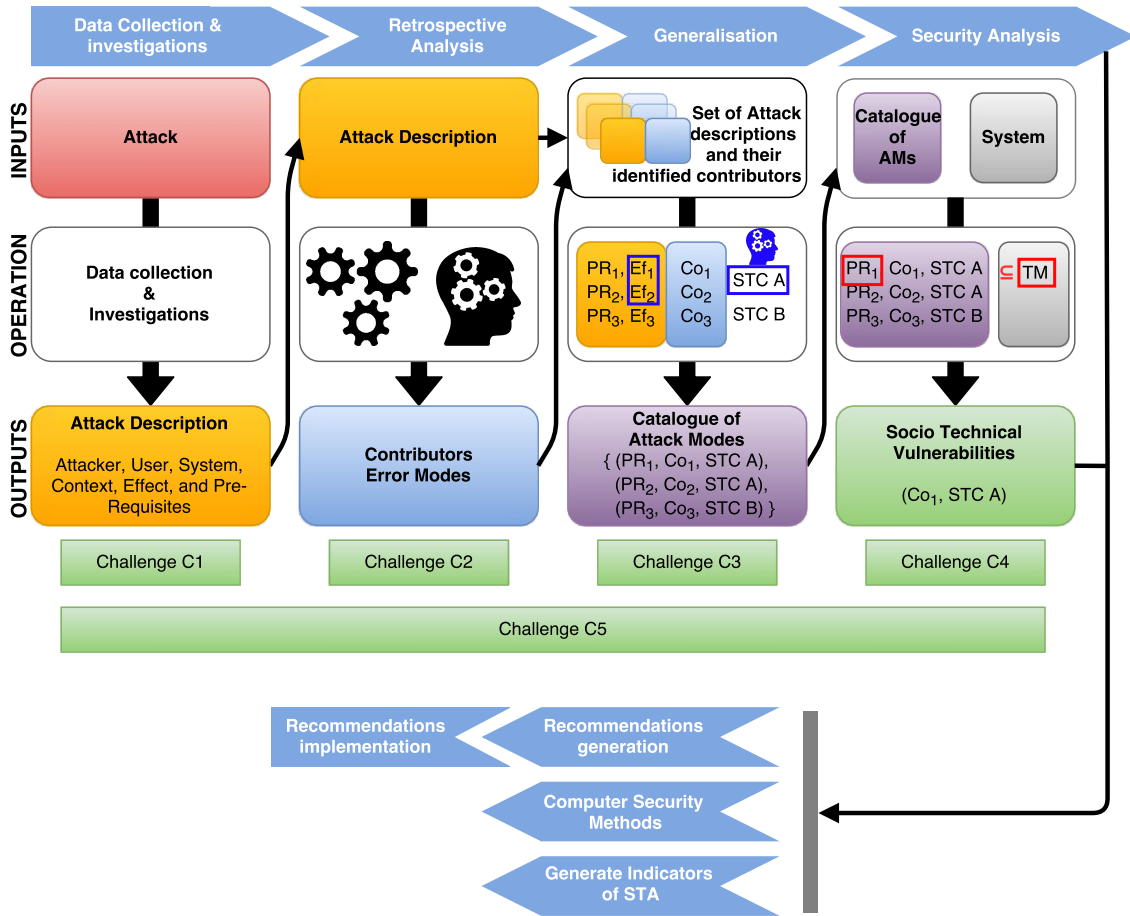


Figure 6.2: S-CREAM process and steps. The figure's *Security Analysis* step (see Section 6.2.7) sketches the semi-automatic modality. An expert-driven modality reuses the *Data Collection and Investigations* and the *Retrospective Analysis* steps to investigate potential attacks, and then appends its products to the semi-automatic *Security Analysis*'s results. Acronyms used in the figure: PR = Pre-Requisites, Co = Contributor, Ef = Effect, STC = Socio-Technical Capability, TM = Threat Model.

### 6.2.4 First step: Data collection, and investigations

This step addresses  $C_1$  and supports the resolution of the other challenges by structuring the data related to the attack under scrutiny.

#### Concepts

**Definition 1 (Attack description scheme)** *The Attack description scheme is a template that guides the collection of information. The scheme contains the list of information needed for the subsequent analysis to be performed successfully.*

**Definition 2 (Pre-Requisites)** *The Pre-Requisites (PRs) are flags associated with the different fields contained in the Attack description scheme. They represent the capabilities that an attacker is required to possess in order to perform the attack.*

## Inputs

- The factual information gathered about the attack (i.e., through Digital Forensics, Incident Response, live monitoring, and in person investigation).
- An attack description scheme. The scheme will define the structure of the information that this and the other steps will process, and the PRs that can be applied on this information. This scheme structures the way the analyst will later on generalise attacks, and test other systems for socio-technical vulnerabilities. For instance, expressing attacks' PRs in terms of capabilities on protocols is different from expressing these same PRs in terms of capabilities on security ceremonies. We take these decisions for our implementation of the *Data Collection and Investigations* steps described in Section 6.3.2. Furthermore, information about the attacker, the effect of the attack on the system, the system, and the context will be used in the *Retrospective Analysis*. This information constrains the universe of Contributors that can be discovered. For instance, Contributors related to the user state of mind cannot be yielded by the *Retrospective Analysis* if not collected and stored in the attack description.

**Output** A factual, structured description of the attack with its PRs.

**Operation** The analyst uses the gathered information to fill the scheme and seeks further information about the attack when needed. Information can be gathered through different means of investigation. The security analyst ponders upon which fields to define as required by setting the PRs on the attack's description (by ticking check boxes for instance).

We now give an example with a generic scheme. This scheme asks for a textual description of the *Attacker's actions*, their *effect on the system's security*, the *User's behaviour*, the *System*, and the *Context*. Here is a description of the output produced by the *Data Collection and Investigations* step using the example scheme, and the toy attack example as inputs:

- Attacker's actions: what information did the attacker gather to perform the attack, what is the form and content of the messages sent, what is their time of appearance *et cetera*. In our example, the attacker sends a message imitating the graphical identity of  $I_1$  and a request to perform  $A_1$  on  $S_1$ .
- Effect on the system's security: what are the consequences of the success of the attack on the system. What the attacker achieved. In our example, the unauthorised attacker performed the action  $A_1$  via the user.
- User's behaviour: what are the user's behaviours that allowed the attack to be a success, what should the user have done to avoid the attack. In our example: the user performs  $A_1$  on  $S_1$ .

- System's description: information about the interactions between the user and the system as well as about the interfaces mediating these interactions. In our example,  $I_1$  is authorised to issue requests to the user.
- Context's description: information about the context in which the attack occurred and the context in which the user-system interactions usually occurs.
- PRs to perform the attack: what the attacker needs to control and what capabilities he needs to be able to perform the attack. In our example, the attacker needs to be able to 'send a message' and to 'imitate the graphical identity'.

We describe our implementation of the *Data Collection and Investigations* step in Section 6.3.

### 6.2.5 Second step: Retrospective Analysis

The *Retrospective Analysis* step addresses the challenge that we called  $C_2$  in Chapter 5. It provides the analyst with a method to extract the Contributors of an attack's success.

#### Concepts

**Definition 3 (Error Mode)** *An Error Mode (EM) is the analogue of a failure mode as is used in technological failure analysis. An EM describes the observable user's erroneous behaviour in space and time. Defining the EM of an observed erroneous action will for instance clarify if the user performs an action at the wrong time, or an action on the wrong object.*

**Definition 4 (Contributor)** *A Contributor is a characteristic pertaining to any component of the system where the attack has occurred, that has facilitated the attack's success.*

**Inputs** It is the output of the *Data Collection and Investigations* step. In particular, it is information about the Attacker's actions, the user's behaviours, the system and the context.

#### Output

- A list of *EMs*, or the errors corresponding to the user's observed behaviour that permitted the success of the attack.
- A list of Contributors that are responsible for the occurrence of the said observed behaviour.

**Operation** This step's operation is based on cause-consequent links that the analyst follows until he identifies likely Contributors to the attack. The analyst first identifies what *EMs* drove the user's insecure behaviour, then he follows the causal relationships S-CREAM provides until he finds a satisfactory list of Contributors.

In our toy example, the observed error mode is that the user misidentifies a message as emanating from  $I_1$ . We reckon that the *Retrospective Analysis* would identify the fact that the received message mimicked the graphical identity of this entity as Contributor. With additional information, S-CREAM could identify more Contributors. For instance, the malicious message could have been received at a time when a genuine message from the spoofed entity was expected, and the ‘Habits and expectations’ Contributor could have been selected.

In the setting of our process, the identified Contributors are used in the next step: the *Generalisation* step. It is worth mentioning here that if one wanted only to study one attack on a particular system, one could follow the generic process of RCA described earlier and produce recommendations to thwart this attack on this particular system. Such recommendations are not directly provided by S-CREAM. It is the analyst’s duty to produce the recommendations about the controls that should be applied to prevent the recurrence of the attack on the system, from the list of Contributors yielded by the *Retrospective Analysis*.

Our implementation based on CREAM along with the associated tool are described in great lengths in Section 6.3.3 and in Section 6.4.

## 6.2.6 Third step: Generalisation

The *Generalisation* step addresses the challenge that we called  $C_3$  in Chapter 5. Unlike the other steps, the *Generalisation* requires not only one but several successful preceding steps (*Retrospective Analysis*) to operate. The *Generalisation* step is meant to create reusable links between the attacker’s capabilities and security incidents. These links then enable an analyst to probe a system for socio-technical vulnerabilities under a Threat Model (see the *Security Analysis* step).

### Concepts

**Definition 5 (Socio-Technical Capability)** *A Socio-Technical Capability (STC) is the capability to produce an effect harmful for a system’s security by performing a user-mediated attack. This effect can be directly or indirectly harmful for the system’s security. This means, producing an effect on another component that will ultimately harm the system’s security is also an STC.*

In our toy example, we can identify the action  $A_1$  as being an STC, however, we can also identify *Identity Spoofing* as an STC because spoofing  $I_1$ ’s identity is an effect on the user that ultimately leads to performing  $A_1$ .

**Definition 6 (Attack Mode)** *An Attack Mode (AM) is a link between an STC, **one** of its Contributors, and some PRs*

In our example attack, if we identify ‘Expectancies of a message’ as a Contributor, then an AM could be built through the following:

- ‘Identity Spoofing’ as STC,
- ‘Expectancies of a message’ as Contributor, and
- ‘Send a message’ and ‘Reproduce any graphics’ as PRs applied on the attacker’s capabilities.

**Definition 7 (Socio-Technical vulnerability)** *A socio-technical vulnerability is the presence of an uncontrolled AM in a system.*

**Definition 8 (Catalogue of Attack Modes)** *A catalogue of AMs is a repository of AMs.*

**Inputs** A set of the outputs obtained from the *Data Collection and Investigations* and *Retrospective Analysis* steps performed on previously studied attacks.

**Output** Attack Modes compiled into a catalogue.

**Operation** The analyst builds Attack Modes by grouping the outputs from several corresponding *Data Collection and Investigations* and *Retrospective Analysis* steps together. The analyst chooses which STCs to create depending on the attacks he analysed before the *Generalisation* step. For instance, if he studied several attacks related to pretexting, he may have gained knowledge about the Contributors that facilitate spoofing identities. From this list of Contributors and descriptions of attacks, he can decide to create a ***Spoof*** STC.

Indeed, generalising a set of attacks into a catalogue of AMs starts with a simple question: ‘What are the STCs gained through these attacks?’. Once this question is answered, the description of the AMs consists of, for each AM, appending the PRs defined in the *Data Collection and Investigations* step and one Contributor identified during the *Retrospective Analysis* step of an attack to the STCs, which the analyst identified has being gained during this attack. The catalogue of AMs is built by repeating this process for all the identified Contributors that correspond to the attacks that the analyst wants to generalise.

For instance, if we represent an AM as a the three constituents: PRs (PR), a Contributor (Co), an STC (STC); an AM built from the attack in our example would be:

$$(PR_1, Co_1, STCA)$$

With:

- $PR_1 =$  (Send message, Imitate graphical identity)
- $Co_1 =$  (Habits and Expectancies)
- $STCA = A_1$

But this AM is not generalised as the action  $A_1$  can be specific to the system  $S_1$  on which it is performed. If we want our AMs to be reusable and helpful in preventing an attacker to perform an attack, say Attack 2, identical to Attack 1 but launched on a different system, and therefore, targeted to the specific action of this second system, we need to set the AM's STC to a common effect between these attacks.

As defined earlier, an STC can be an effect that ultimately causes harms to the system's security. We can thus choose to build AMs with the STC 'Identity Spoofing' and obtain:

$$(PR_1, Co_1, \textit{Identity spoofing})$$

$$(PR_2, Co_2, \textit{Identity spoofing})$$

The AMs now reflect what has been observed in past attacks and provide links between an Attacker's capabilities, effects that have had security consequences in the past, and what Contributors one should control to prevent this kind of attack from recurring. Furthermore, Contributors can be used as *Indicators* of an STA as the manipulation of these Contributors can betray the occurrence of an attack.

Our implementation of the *Generalisation* step is described in Section 6.3.4.

## 6.2.7 Fourth step: Security Analysis

This step is where the information gained in the *Retrospective Analysis* step and generalised in the *Generalisation* step can be reused by the analyst for addressing the challenge  $C_4$  introduced in Chapter 5. This step operates following the two following modalities: a mandatory semi-automated *Security Analysis* and an optional expert-driven *Security Analysis*.

### 6.2.7.1 Semi-automatic Security Analysis

In this step's modality, the analyst makes use of a catalogue of AMs previously built by generalising STAs. The analysis identifies socio-technical vulnerabilities in a system by filtering the catalogue of AMs by the Threat Model that applies to this system.

#### Concepts

**Definition 9 (Threat Model)** *Determines the attacker's capabilities, and optionally, the attacker's goal in terms of the STC in question.*

For instance, an attacker can be enabled to send messages optionally with the goal of spoofing identities.



**Definition 10 (Contributor)** *In this prospective context, a Contributor is a characteristic pertaining to any component of the system that can contribute (because it already did in the past on another system in another attack) to an attack.*

**Inputs** The inputs are a catalogue of AMs and the system to be analysed.

**Output** This step outputs a list of socio-technical vulnerabilities. That is to say a list of STCs, Contributors couples.

**Operation** The analyst must, in turn, perform several operations, which are as follows:

First, the analyst describes the Threat Model that is applied to the system in the same scheme used to describe the attacks employed to build the catalogue of AMs.

Then, the analyst filters the catalogue of AMs in order to list the AMs which have PRs that fit into the system's Threat Model.

Hence, if the catalogue is only made of the two AMs built from the previous example and that for the Threat Model  $TM_S$ ,  $PR_1$  does not fit  $TM_S$  and  $PR_2$  fits  $TM_S$ :

$$\neg(PR_1 \subseteq TM_S) \wedge (PR_2 \subseteq TM_S)$$

Then, only the AM with the PRs  $PR_2$  is listed in the output. It constitutes the list of potential socio-technical vulnerabilities identified for the system under the Threat Model  $TM_S$ :

$$(Co_2, Identity\ spoofing)$$

### 6.2.7.2 Analyst-driven Security Analysis

The analyst-driven *Security Analysis* is an optional path for the *Security Analysis* where the analyst can input additional insights into the results.

#### Concepts

**Definition 11 (Potential Attack)** *An attack that the analyst reckons is possible against the system but for which no factual information exists. It is a plausible 'what if' scenario.*

**Inputs** The knowledge the analyst has of the potential attack.

**Output** A list of Contributors to the potential attack.

**Operation** The analyst must, in turn, perform several operations, which are as follows:

First, the analyst identifies a potential attack against the system under scrutiny.

Then, he proceeds to investigate this attack by performing a *Retrospective Analysis* step (preceded by a *Data Collection and Investigations*), which yields the attack's Contributors.

Our implementation of the *Security Analysis* step is described in Section 6.3.6.

## 6.2.8 Controlling socio-technical vulnerabilities

S-CREAM's outputs were designed to allow an analyst to thwart potential user-mediated attacks by: (i) applying controls on the Contributors of identified socio-technical vulnerabilities, (ii) leverage the use of computer security methods by listing the STCs attainable by an attacker given a Threat Model, and (iii) providing *Indicators* of STAs.

Thus, depending on the presence of a Contributor in the system and the success of the different methods, there are several paths that an analyst can follow:

- The Contributor is not found: In this case, the socio-technical vulnerability does not exist in the system.
- The Contributor is found and reliable controls can be applied: In this case, the analyst applies recommendations as he would have done in a generic RCA process. The socio-technical vulnerability is controlled and the system's security is safe.
- The Contributor is found but no reliable control can be applied: In this case, the system has a gaping socio-technical vulnerability. The analyst can then turn to computer security methods to prove that the system is secure against an extended Threat Model that incorporates the newly discovered socio-technical vulnerability. If it can't be proven secure, then the analyst can attempt to redesign the system to make it secure against the extended Threat Model.
- If everything else fails, S-CREAM's outputs can be considered to create sophisticated *Indicators* of STAs that can be used, for instance, to monitor the system and respond quickly to a security incident.

## 6.3 S-CREAM's implementation

We describe our implementation of S-CREAM, the technique we devised by customising the Cognitive Reliability and Error Analysis Method (CREAM) and that proposes as a way to identify Contributors of Socio-Technical Attacks (STAs).

### 6.3.1 Implementation Choices

#### 6.3.1.1 Building the Retrospective Analysis at the heart of the Root Cause Analysis method

We selected CREAM [91] as our preferred Root Cause Analysis (RCA). CREAM is a 2nd generation Human Reliability Analysis (HRA) method. By considering cognitive causes of errors, CREAM brings a great deal of details into the analysis of an accident, and because of such richness in details, it has been criticised in HRA [27]. However, such richness is what makes CREAM a great candidate for computer security. Indeed, more than identifying solely the root cause of an attack, we are interested in identifying its potential Contributors, or in other words, all the factors that an attacker could have used to push the human involved in the attack to behave erroneously and compromise the system's security. Among other criteria, the most important aspect of CREAM is that it offers retrospective and prospective analysis. Thus, it provides us with bi-directional links between causes and effects. This allows us to build a catalogue of Attack Modes (AMs) that can be used for both detecting attacks (starting from observed effects) and predicting attacks (starting from a threat model).

CREAM relies on the two following pillars: (i) a classification of erroneous actions (this is represented in tables linked together by causal relationships) and (ii) a method that describes how to follow these links back to the human as well as the contextual and the technological factors at the origin of an 'event'. An event is caused by the manifestation of an 'erroneous action', and is called the phenotype [91]. The confluence of underlying factors that made the erroneous action arise is called its genotype. CREAM's tables of causal relationships between antecedent (cause of errors) and consequent (effect of errors) link a phenotype with its genotype [91]. Following these causal relationships, it is possible to find what caused an erroneous action and the root cause(s) of an event.

CREAM is the building block of our method, however, it needs to be customised for security. We call the result S-CREAM, which stands for 'Security CREAM'. S-CREAM is explained in the following sections.

#### 6.3.1.2 Choosing a source to bootstrap S-CREAM's catalogue of AMs

We bootstrap S-CREAM's catalogue of AMs with a library of known attack patterns drawn from Common Attack Pattern Enumeration and Classification (CAPEC) [41]. This library contains attacks 'generated from in-depth analysis of specific real-world exploit examples.'<sup>1</sup> It is maintained by MITRE Corporation, and it compiles and documents a wide range of attacks centered on the user. We use CAPEC's repository to extract and select these Attack Patterns whose success relies on critical actions of the user. The CAPEC taxonomy contains descriptions of social-engineering Attack Patterns, together with their Pre-Requisites (PRs), mechanisms, and possible mitigations.

---

<sup>1</sup>See <https://capec.mitre.org/>

### 6.3.2 Implementing the *Data Collection, and Investigations* step of S-CREAM

There are two aspects of importance when implementing the *Data Collection and Investigations* step described in Section 6.2: (i) the implementation should enable the analyst to perform a sub-sequent *Retrospective Analysis* step to objectively choose the paths to follow through antecedent-consequent links, and (ii) the implementation should structure the information about the attacker's capabilities required to perform the attack in a way that allows the analyst to filter the attacks by these capabilities.

It is the attack description scheme that defines what data are to be collected, and how the data should be organised, i.e., what attack properties should be defined. The scheme is customisable. We choose to describe the effects the attack has on a system's security and the attacker's actions.

**Describing the effects.** We choose to describe the consequences of an attack on the user and on the system in text because a precise description is the key to a successful *Retrospective Analysis*. Indeed, there are so many applications, user interactions, decision processes, and consecutive actions possible, that the text is the best way to forward a fine-grained description of an attack. Thus, this implementation deviates from the *Data Collection and Investigations* step described in Section 6.2 by lacking structure to describe the effect an attack produces on a system. This is because creating a proper scheme is a complex matter that will be addressed in future works, as discussed in Chapter 9.

**Describing the attacker's actions.** In contrast to the effects, we describe the attacker's actions in a very structured way. We follow what has been proposed when introducing STEAL in Chapter 3, i.e., describing attacks as a set of messages flowing between the attacker and the victim prior to the manifestation of the critical action. With the notable distinction that we don't use UML diagrams to describe all the messages flowing between the user and the attacker, we focus on the attacker's messages, and we extract from them a set of common properties. Thus, the event that initiates the attack is described through common properties shared by the messages sent from the attacker to the user. These properties are as follows:

1. a source, which is the principal that the user believes to be interacting with,
2. an identity split into declared identity (i.e., who the attacker says he is, like the from field of an email) and imitated identity (i.e., who the attacker imitates to be by stealing a logo for instance),
3. a command for the user to execute,
4. a description of the subsequent action to state if the action is booby-trapped or spoofed,
5. a sequence that describes the temporal situation of the message, and
6. a medium (web, phone, or paper).

**Describing the PRs.** To describe the attacker's capabilities that are needed to perform the attack, we attach a *pre-requisite* flag to each property contained in the scheme.

### 6.3.3 Implementing the *Retrospective Analysis* of S-CREAM

S-CREAM's *Retrospective Analysis* draws from the retrospective analysis of CREAM and includes adaptations required by our computer security focus. We sketch CREAM's original analysis, and S-CREAM's analysis side-by-side in Figure 6.3.

In CREAM's retrospective analysis, one first defines common performance conditions (see left-hand side of Figure 6.3) to describe the analysed event followed by the Error Modes to investigate. This investigation is a process where the analyst searches for the antecedents of each Error Mode. This process is recursive, i.e., each antecedent that an analyst finds can be investigated in turn. Antecedents justified by other antecedents are called 'generic', whereas those which are 'sufficient in themselves' are called 'specific'. To avoid following 'generic antecedents' endlessly, one must stop the investigation on the current branch when a 'specific antecedent' is found to be the most likely cause of the event (in Figure 6.3, see the *yes* branch on the right-hand side of the CREAM block that leads to the end state for the current branch).

As discussed in ( $D_6$ ) (see Chapter 5), the computer security context in which we intend to use CREAM's retrospective analysis calls for a different procedure. Two adaptations of CREAM's retrospective analysis method are therefore needed.

First, we customise the phase preceding the investigation instead of formalising the context in common performance conditions. S-CREAM implements a *Data Collection and Investigations* step as described in the previous section (in Figure 6.3, the description step depicted as a green box replaces the first activity of CREAM).

Second, S-CREAM uses a less restrictive stop rule to yield Contributors. By doing so, we avoid pointing invariably to the attacker's action, which allow us to investigate additional contributing antecedents. Hence, where CREAM stops as soon as a specific antecedent is found as being a likely cause of the event, S-CREAM lists all likely specific antecedents for the event in addition to the specific antecedents that are contained into sibling generic antecedents. It then stops the investigation of the current branch.

As S-CREAM's *Retrospective Analysis* follows a *Data Collection and Investigations* step (see also the red box in Figure 6.3), the analyst uses the description of the attack to define the critical actions carried out by the victim (i.e., those with an effect on the system's security) and the associated Error Modes. For the *Retrospective Analysis* to be possible, the analyst has to identify at least one Error Mode for an attack. Additional Error Modes may have to be analysed in the course of the events that lead to the critical action, for instance, if the victim first encounters the attacker and misidentifies him/her as being trustworthy. Considering each antecedent with the attack's description in hand, the analyst follows the stop rule to build the list of Contributors of the attack under scrutiny.

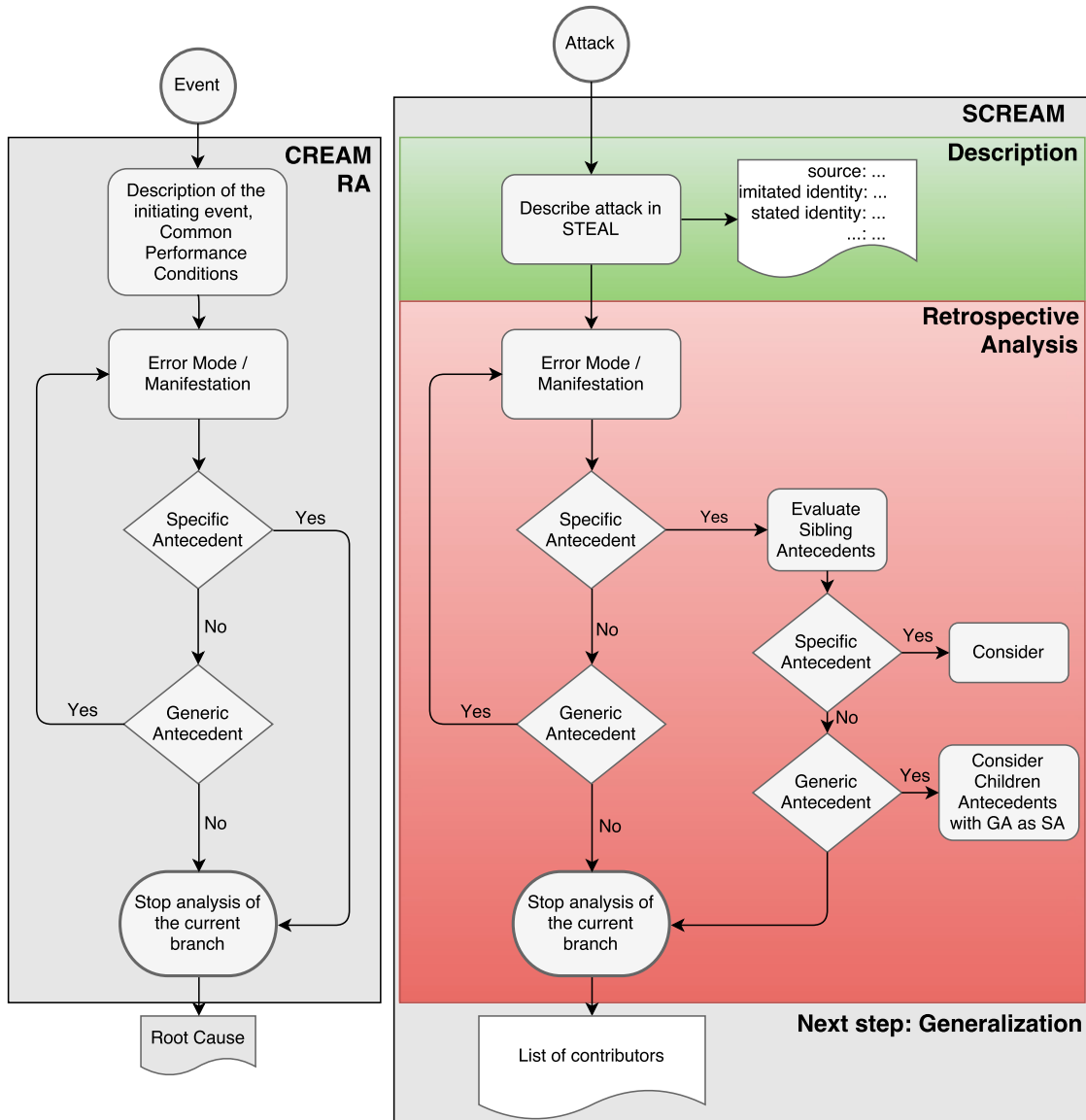


Figure 6.3: Side-by-side comparison between CREAM (left-hand side) and S-CREAM (right-hand side) processes.

### 6.3.4 Implementing the *Generalisation* step of S-CREAM

**Building Socio-Technical Capabilities.** The *Generalisation* step’s description in Section 6.2 states that this step builds AMs, or links between an attacker’s capabilities and the effects he can potentially produce on a system’s security. For instance, sending a message that nudges a user to click on a malicious link may allow the attacker to execute some code on the system. There may be an observable instance of such an attack that exists, but it is not reasonable to create a Socio-Technical Capability (STC) called ‘Remote Code Execution’ (and with it all the AMs with the Contributors that make it possible) that links an attacker that can send a message with this STC. Indeed, there are so many differences in the usage and the consequences of this action of clicking that it makes no sense to create STCs based on its consequences, instead creating an STC ‘Push the user to click on an ob-

ject' would make more sense. For the initial development of S-CREAM's catalogue of AMs, we are interested in STCs that are the intermediate goals of the attacker and are peripheral and decoupled from the system on which they are exploited.

The STCs that S-CREAM will initially provide are links between what the attacker can do and the consequences on the **user** and not the **system**. We gear S-CREAM's *Generalisation* step towards STCs related to the user. Here is a list of STCs that we foresee could be reachable by the use of S-CREAM (given that the attacks that make use of these capabilities exist in the corpus of attacks used to build a catalogue) and actually useful for computer security methods:

- **Spoof** The attacker is able to usurp another entity's identity. This STC is, for instance, used in phishing to impersonate an entity that is likely to send the request contained in the phishing email.
- **Block** The attacker is able to block messages reaching the user by distracting him for instance.
- **Alter** Some AMs may provide the attacker with the capability of changing the perception the user has of a genuine message.

**S-CREAM allows to hide some Attack Modes.** As for STCs, Contributors can be highly specific to a system or a context of attack. Thus S-CREAM provides a way (a flag) to remove AMs that use a Contributor that is highly unlikely to be found in other systems than the one in which it was observed (sight parallax-related Contributors for instance). This flag can be set at any time, thus buying time for the analyst to judge the usefulness of an AM while performing *Socio-Technical Security Analyses* before deciding on its filtering.

The *Generalisation* step can be run numerous times on diverse sets of attacks to enrich the resulting catalogue of AMs (and nothing forbids the creation of different, specialised catalogues). In the next section, we bootstrap the catalogue using CAPEC as the attack corpus.

### 6.3.5 Bootstrapping S-CREAM's catalogue of Attack Modes

As explained in Section 6.3.1.2, we now bootstrap S-CREAM's catalogue of AMs yielding attacks from the Common Attack Pattern Enumeration and Classification (CAPEC). Building a catalogue of AMs consists of: (i) Investigating a corpus of attacks through the use of the *Data Collection and Investigations* and *Retrospective Analysis* steps, and (ii) generalising the results.

#### 6.3.5.1 Investigating CAPEC's attacks

We identified 15 Attack Patterns out of CAPEC (retrieved on January 2016) where the user is at the source of the success of the attack. Attack Patterns (APs) children of *CAPEC-403 Social Engineering* were not selected in the first place because they did not provide enough details about the messages flowing between the attacker and

its victim or the effect on the system's security. In addition, *CAPEC-485: Signature Spoofing by Key Recreation* was ignored because one of its PRs is that the attacker can break private keys, and we usually consider unbreakable private keys in security analysis.

The list of the analysed APs along with the identified Contributors is shown in Table 6.1. For the purpose of illustration, we detail the analysis of one Attack Pattern, the CAPEC-195 'Principal Spoofing'. This Attack Pattern is not considered an issue solely from technical point of view. Its root cause mostly depends on the user's weaknesses and technical factors further increase its likelihood. We first detail how we integrate this Attack Pattern into our framework, then we perform a S-CREAM analysis of its causes of success.

**Description step of CAPEC-195 'Principal Spoofing'.** In the CAPEC-195 'Principal Spoofing' Attack Pattern, the attacker pretends to be an additional actor in the interaction. This attack relies on the perception that the message content has been sent by an honest identity. Its description in our framework can be summed up by the following: (1) the source is another principal that the target knows, (2) the imitated identity is used because the appearance of the message is crafted to reflect the source's identity, (3) the command is not specified, (4) the attacker is the initiator of the non-spoofed action 'disclose information' or 'perform action on behalf of the attacker', (5) the message is a continuation of a previous interaction as the target must know the principal, and (6) the medium can either be in person, or through the phone, smart phone, operating system, Wifi, paper, or web.

**Retrospective Analysis of CAPEC-195 'Principal Spoofing'.** The main Error Mode (EM) of this Attack Pattern is the misidentification of the attacker for another principal. We identify this as being a 'Wrong object:Similar Object' EM. Figure 6.3 shows which path we follow among the possible antecedents for this EM. As the declared identity is not used in this attack, the specific antecedents related to the labeling do not contribute to the behaviour. Therefore, we continue the analysis by looking at the generic antecedents, and following the generic antecedent 'Wrong identification:Incorrect identification', the specific antecedent 'Erroneous information' is selected as Contributor because the imitated identity is spoofed. This root cause provided by S-CREAM is the same as the explanation provided by CAPEC: the wrong information provided by the attacker tricks the user. As shown in Table 6.2, we follow our custom stop rule and consider the other specific antecedents and sibling generic antecedents for this branch.

### 6.3.5.2 Generalizing the results into a catalogue of AMs

Starting from the results of the 15 CAPEC's APs analysed with S-CREAM, we build a catalogue of AMs that can further be used when analysing other systems (see Table 6.1).



Table 6.1: Table listing the AMs corresponding to the *Identity Spoofing* and the *Action Spoofing* STCs. The right column displays the Contributors that the AP allowed us to identify as linked to the STC. Contributors are only appended once to the list of each STC's Contributors and other components of AMs such as PRs are not displayed for the sake of space. APs are sorted numerically.

STCs	CAPEC Attack Patterns	List of Contributors
<i>Identity Spoofing</i>	89 - Pharming	SA: Incorrect label SA: Presentation failure SA: Ambiguous symbol set SA: Ambiguous signals SA: Erroneous information SA: Habit, Expectancy SA: Inadequate training SA: Multiple signals GA: Missing information GA: Faulty diagnosis GA: Mislabelling GA: Wrong reasoning GA: Insufficient knowledge
	98 - Phishing	SA: Too short planning horizon
	163 - Spear Phishing	
	164 - Mobile Phishing	
	194 - Fake Source of Data	
	195 - Principal Spoof	SA: Competing task SA: Mis learning SA: Error in mental model
	476 - Signature Spoof by Misrepresentation	GA: Inadequate quality control GA: Inattention
	477 - Signature Spoof by Mixing Content	
	103 - Click jacking	SA: Incorrect label SA: Presentation failure SA: Erroneous information SA: Habit, expectancy
	181 - Flash File Overlay	SA: Inadequate training SA: Too short planing horizon SA: Insufficient knowledge
<i>action spoofing</i>	222 - Iframe Overlay	
	501 - Activity Hijack	
	504 - Task impersonation	
	505 - Scheme Squatting	
	506 - Tap Jacking	SA: Ambiguous symbol set SA: Ambiguous signals SA: Multiple signals

Table 6.2: Justifications for the selections of Contributors for the ‘Wrong identification’ generic antecedent. Specific antecedents inside generic antecedents are not displayed and specific antecedents and generic antecedents are abbreviated as SA and GA.

Antecedent	Justification
SA ‘Ambiguous Signals and Symbols’	The usability of the interface can contribute to this Error Mode.
SA ‘Habit and Expectancies’	As the message sent by the attacker is a continuation from a previous interaction we can reasonably consider that this antecedent plays a role in the target’s behaviour.
GA ‘Distraction’	We don’t have additional information regarding the SAs contained in this GA in our description. But it is likely that the user was performing a main task while assessing the identity of the attacker, so we consider SA ‘Competing task’ as an additional Contributor.
GA ‘Missing Information’	The attacker deliberately hides its real identity and the presentation fails to clearly state the sender identity. So we consider the corresponding SAs as Contributors.
GA ‘Faulty Diagnosis’	The user may have a wrong mental model about how to assess identity, or misunderstood previous explanations.

### 6.3.5.3 Choosing the *Socio-Technical Capabilities*

To decide what STCs we add in the catalogue of AMs, we take inspiration from CAPEC’s hierarchical structure. Indeed, it appears that all the studied APs were pulled from two meta APs: *CAPEC-151: Identity spoofing* and *CAPEC-173: Action spoofing*, and hence, we chose to create two STCs which are ***Identity spoofing*** and ***Action spoofing***:

- ***Identity spoofing*** : The attacker is able to usurp an identity. From a user standpoint, the attacker is no different from the spoofed source.
- ***Action spoofing*** : The attacker is able to change an action’s target. The user is deceived into thinking that an action he performs will behave as he expects, whereas another action, which is harmful for the system, is executed in its place. Opening a SMS that bricks a smart phone is an instance of such spoofed action.

We detail the *Generalisation* step for one STC built from a reduced set of two APs, then we give an overview of the full catalogue.

### 6.3.5.4 Compiling APs’analyses results into *STCs*

Despite being built from a small set of APs, the catalogue of AMs that we built cannot be listed in this document. To illustrate the *Generalisation* step, we work with the results of two APs that we compile into one STC. We focus on the results from the S-CREAM analyses of CAPEC-195: ‘Principal Spoofing’ and CAPEC-194 ‘Fake source of data’.

Table 6.3 compiles this reduced list of AMs where we can see the PRs that an attacker needs in order to gain the ***Identity spoofing*** STC against a system. As explained in Section 6.3.4, we only consider PRs related to the attacker in our implementation. Table 6.3 displays on its left-hand side the Contributors yielded from the analyses of the APs, and on its right-hand side, the PRs an attacker needs

Table 6.3: Reduced list of *AMs* for the ***Identity spoofing STC***. The attacker's capabilities have to match the set of the *AM*'s PRs in order for the attacker to be able to use the *AM*. For instance (see highlighted row), an attacker has to be in control of his 'Declared identity' (he can declare himself as anyone) in a message that is a continuation of a previous interaction between the victim and the source (on any medium) to gain the ***Identity spoofing STC*** by exploiting the 'Incorrect label' Contributor. *AMs* that only require the attacker to be able to send a message are not shown for the sake of space, namely: *Bad mental Models*, *Mislearning*, and *Multiple Signals*.

Attack Modes Contributor	Attacker pre-requisites						
	Source	Declared Identity	Imitated Identity	Command	Action	Sequence	Medium
Ambiguous symbols	-	-	<i>rq</i>	-	-	<i>rq</i> : continuation	<i>rq</i> :any
<b>Incorrect label</b>	-	<i>rq</i>	-	-	-	<i>rq</i> : continuation	<i>rq</i> :any
Erroneous Information	-	<i>rq</i>	-	-	-	<i>rq</i> : continuation	<i>rq</i> :any
	-	-	<i>rq</i>	-	-	<i>rq</i> : continuation	<i>rq</i> :any
Habits and Expectancies	-	<i>rq</i>	-	-	-	<i>rq</i> : continuation	<i>rq</i> :any
	-	-	<i>rq</i>	-	-	<i>rq</i> : continuation	<i>rq</i> :any
Competing task	-	<i>rq</i>	-	-	-	<i>rq</i> : continuation	<i>rq</i> :any
	-	-	<i>rq</i>	-	-	<i>rq</i> : continuation	<i>rq</i> :any
Hidden information	-	<i>rq</i>	-	-	-	<i>rq</i> : continuation	<i>rq</i> :any
	-	-	<i>rq</i>	-	-	<i>rq</i> : continuation	<i>rq</i> :any
Presentation failure	-	<i>rq</i>	-	-	-	<i>rq</i> : continuation	<i>rq</i> :any
	-	-	<i>rq</i>	-	-	<i>rq</i> : continuation	<i>rq</i> :any

to meet for these *AMs* to be usable by an attacker. Some Contributors (i.e., Habit and Expectancies) are duplicated because they appear in both attacks with different sets of PRs.

### 6.3.5.5 The resulting catalogue of *Attack Modes*

As shown in Table 6.1, each of the 15 *APs* has been attributed to corresponding *STCs*. Eight were assigned to ***Identity spoofing*** and seven to ***Action spoofing***. Table 6.1 has been shortened to avoid repetition and does not display the PRs associated with each *AM*.

This catalogue of *AMs* represents the list of Contributors an attacker can exploit to attempt to gain the corresponding *STC* (if its capabilities meet the PRs). We further develop the use of such a construct in the following section.

### 6.3.6 Implementing the *Security Analysis* of S-CREAM

Once the catalogue of *AMs* is bootstrapped, the implementation of this step is only a matter of filtering *AMs* and displaying the Contributors of linked potential attacks. To perform the semi-automated *Security Analysis*, the analyst describes the attacker's capabilities on the system under scrutiny (the Threat Model), and the *S-CREAM assistant* will then filter and display the corresponding *AMs*. To

add his insights about potential attack while performing an analyst-driven *Security Analysis*, the analyst attaches attacks to the system being analysed, as he would attach attacks to an STC. To ensure that the potential attacks are indeed possible on the system, *S-CREAM assistant* filters out the attacks that exceed the attacker's capabilities described during the semi-automatic *Security Analysis*. Albeit being potential attacks, these attacks are no different from regular attacks to the *S-CREAM assistant* and are investigated in the same way (*Data Collection and Investigations* step then *Retrospective Analysis* step). Contributors corresponding to the linked potential attacks are displayed along with the AMs resulting from the semi-automatic *Security Analysis*. The difference is that the AMs instructs the analyst about STCs, whereas the Contributors yielded from the analyst-driven *Security Analysis* provide information on how an attacker could achieve the potential attack's adverse effect on the system.

## 6.4 S-CREAM's companion tool: *S-CREAM assistant*

CREAM is based on tables that the analyst browses through by following causal relationships. This task can be cumbersome and requires the analyst to take several decisions that can undermine the analysis' validity. For instance, the analyst can miss a link to a table or overlook an antecedent that could have been a major Contributor to the success of an attack. To partially address this obstacle, in [67], we used Serwy's software implementation of CREAM [154]. While being perfectly suitable to perform CREAM analyses, this software does not implement any of the customisations described in Section 6.3. We thus had to run and document these parts using pencil and paper. The quality assurance of the results, the low costs of operation, and the usability of the method are the main rationale behind the inception of this tool. *S-CREAM assistant* greatly improves the analysis experience by providing guidance, checking for the completion of the analysis, and providing an interface for navigating and filtering the resulting catalogue of AMs. Indeed, given the number of AMs and the possible filters, it is potentially unbearable to perform some analyses without the support of a dedicated tool. Furthermore, *S-CREAM assistant* facilitates the collaboration of a group of analysts.

*S-CREAM assistant* is an application written in JavaScript that runs locally within a web-browser. Its code is open sourced [95] under the MIT licence.

### 6.4.1 Main functions

*S-CREAM assistant* stores the analyst's work, provides guidance, and automates tedious tasks encountered while running S-CREAM analyses. Its main functions are mapped on the steps described in Section 6.2 and meet the specifications of S-CREAM described in Section 6.3.

**Description.** In *S-CREAM assistant*, the *Data Collection and Investigations* step is accessed from the 'Retrospective Analysis' tab. *S-CREAM assistant* implements

[SCREAM assistant](#)
Retrospective Analysis
Catalogue of Attack Modes
Security analysis

Manage Attacks
Perform the RCA of an attack
View Results

### List of Error Modes identified in CAPEC-195: Principal Spoof

Name	Description	Completed	Action
Wrong object	Similar object	✓	<a href="#">Analyze</a>

[Manage Error Modes](#)
[Compile results](#)

### SCREAM analysis of Wrong object : Similar object

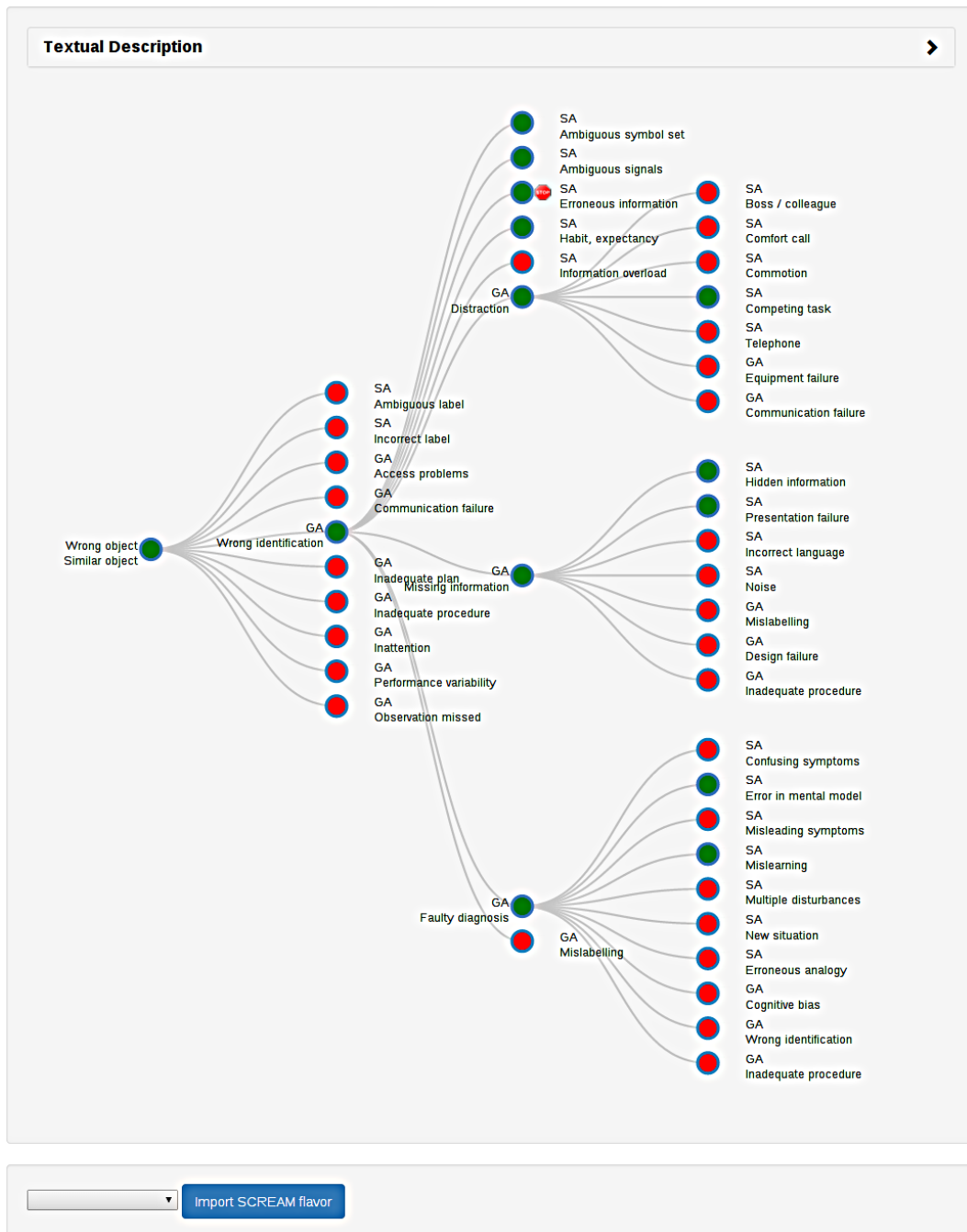


Figure 6.3: (Previous page) Screen capture of *S-CREAM assistant* while performing the *Retrospective Analysis* of the ‘Wrong Object’ Error Mode (EM) observed in CAPEC-195. A green light means that we consider the specific antecedent to be a Contributor or that we expand the generic antecedent. A red light means that we do not consider that the antecedent contributes to the EM. Stop signs show where stop rules are engaged.

the *Data Collection and Investigations* step as a form where the analyst fills in the information related to the attack under scrutiny for the structured description.

**Retrospective Analysis.** As shown in Figure 6.3, *S-CREAM assistant* implements the *Retrospective Analysis* step. The analyst determines the list of EMs observed in the attack under scrutiny from CREAM’s taxonomy of EMs. This is done via a dialog box triggered by the ‘Manage Error Modes’ button. Then *S-CREAM assistant* enables the analyst to investigate each EM with the help of an interactive tree. By clicking on the ‘Analyse’ button, *S-CREAM assistant* pushes the corresponding EM from the list into the tree view for analysis. For an EM under scrutiny, *S-CREAM assistant* displays the possible antecedents that the analyst has to consider as possible Contributors in the form of children to the EM root node. *S-CREAM assistant* takes care of finding the children of *generic antecedents* in CREAM’s tables and of implementing S-CREAM’s custom stop-rule. When a *specific antecedent* is selected or unselected, the corresponding branch is automatically checked to enable the stop rule on the right *specific antecedent* (if applicable), and to open or to close the *generic antecedents* that fall under its realm. That is to say, the *generic antecedents* that are siblings or children of siblings of the *specific antecedent* that carries the stop rule. In addition to this, *S-CREAM assistant* verifies automatically if the tree has reached a ‘completed’ state, meaning that at least one Contributor to the EM under investigation has been found. At any time of this process, the analyst can consult the results of the analysis that are compiled in a dedicated view.

**Generalisation.** To build the catalogue of AMs, *S-CREAM assistant* enables the analysts to store a list of STCs that they can link to the analysed attacks. As shown in Figure 6.4, the tool displays a list of available attacks on the left and a list of linked attacks on the right. An attack can only be linked to one STC at a time (see data model on Figure 6.5 and specifications of S-CREAM in Section 6.2.6). Once the attacks have been linked to an STC, the analyst can compile a list of AMs for this STC. Each AM displays a Contributor, the exploited EM, a justification, which is a comment that the analyst can use to justify his choice, and a check box. The check box allows the analyst to discard an AM from further *Security Analysis* steps by enabling the ‘specific’ flag (as explained in 6.3.4). The view responsible for displaying the AMs hides the details of the description of each AM for the sake of space, while the link to its description exists and is used in the *Security Analysis* step (see below).

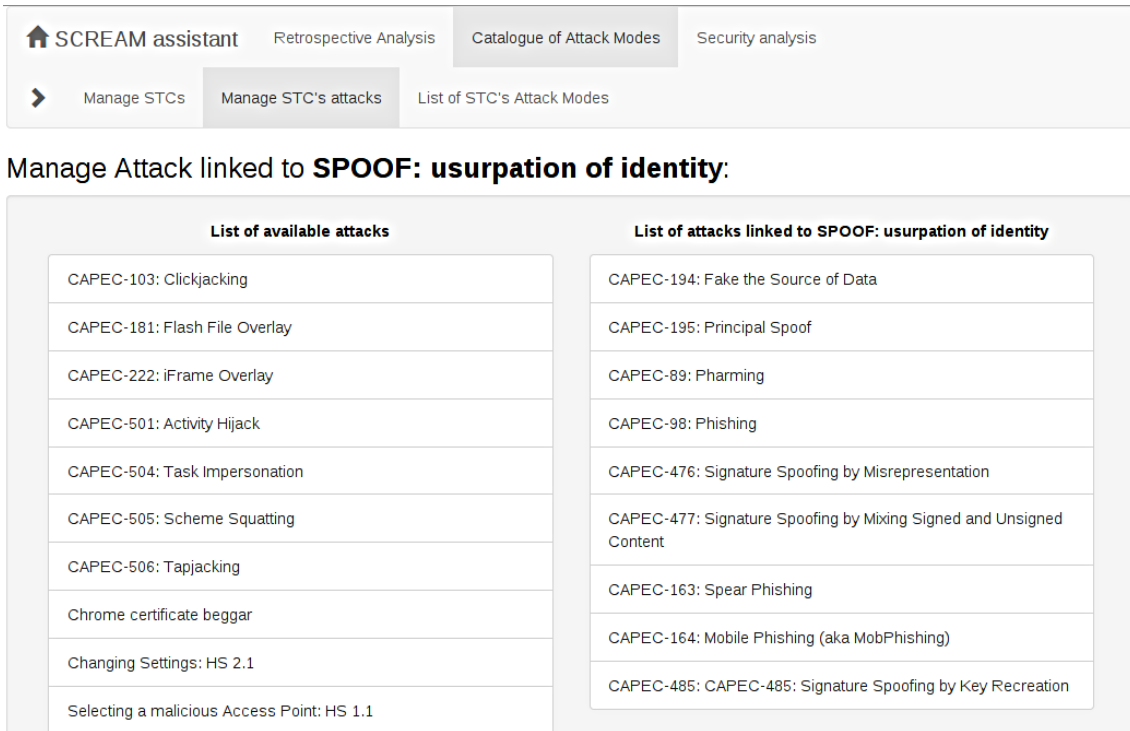


Figure 6.4: Screen capture of the attacker manager for STCs in *S-CREAM assistant*.

**Security Analysis.** *S-CREAM assistant* supports the *Security Analysis* step by allowing the analyst to filter the catalogue of AMs to view only the STCs that fit a given Threat Model. This means that the analyst builds the list of STCs available to an attacker that possesses the Threat Model's capabilities. These capabilities are described in the same way as the attacks are with each property being attached to pre-requisite flag. The list of STCs is built by comparing for each STC, the PRs attached to its AMs with the Threat Model's PRs. An STC is displayed to the analyst only if at least one of its AM fits the system's Threat Model. Finally, the analyst browses the STCs along with their usable AMs corresponding to the system he described.

In addition to listing the STCs along with their AMs corresponding to a system, *S-CREAM assistant* provides the analyst with the possibility to attach attacks to a system (see Figure 6.5). This is meant to investigate potential attacks on a system and get a more precise understanding of its socio-technical vulnerabilities. To add attacks on a system, the analyst uses the same attacker manager as described earlier (see Figure 6.4). The analyst can only select attacks that fit the Threat Model described for the system.

Once, the Threat Model is described and potential attacks are linked, *S-CREAM assistant* displays the list of corresponding STCs (with their AMs) along with the Contributors corresponding to the linked attacks.

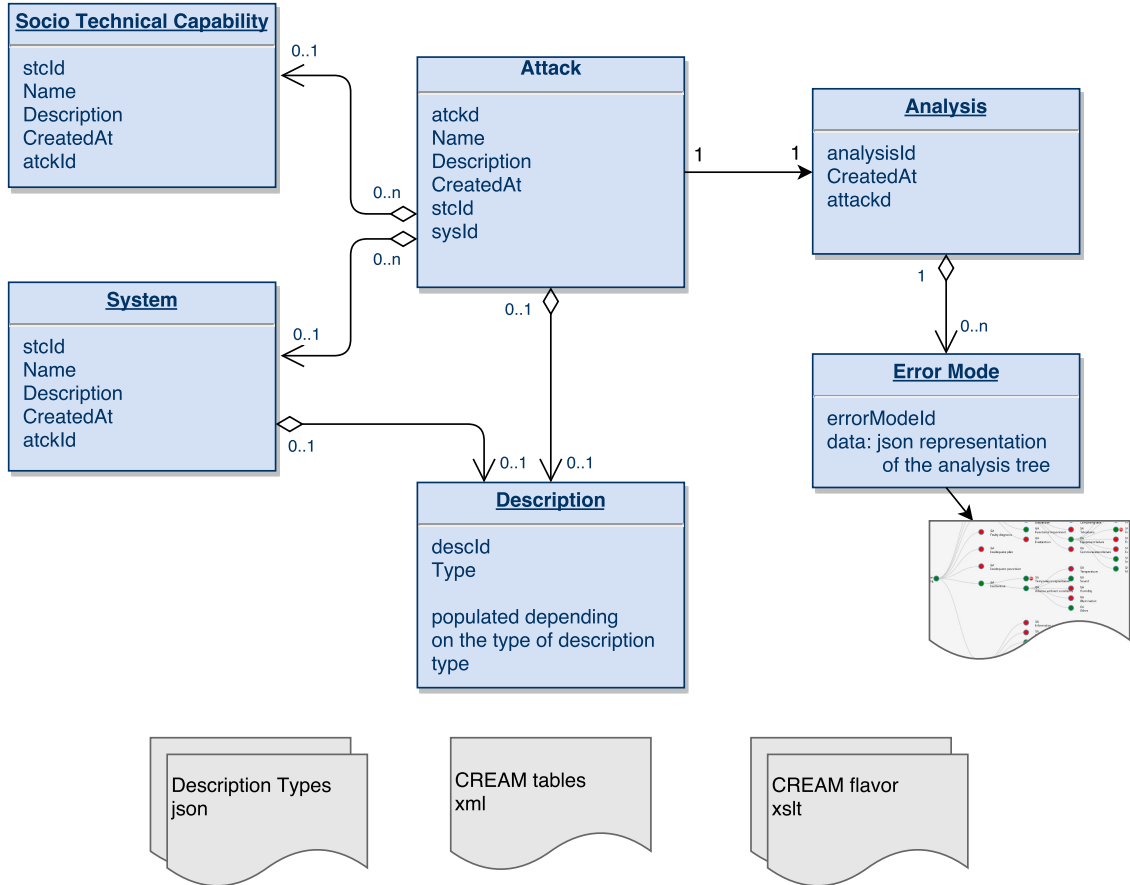


Figure 6.5: *S-CREAM Assistant* data model. Tables in blue are implemented using js-data and stored in the web-browser’s Local Storage.

### 6.4.2 Additional features

*S-CREAM assistant* allows the analyst to export and import the data stored locally. Furthermore, to address  $C_5$  introduced in Chapter 5 (the need for flexibility of the method), and make *S-CREAM assistant* ready to follow further developments of S-CREAM, XSLT style sheets are applied on CREAM tables’ XML representation at runtime. This allows the analyst to add, alter, or remove antecedent-consequent links from S-CREAM in case he performs domain-specific analyses. While performing a *Retrospective Analysis*, the analyst can choose one of these style sheets, which we refer to as S-CREAM *flavors*, to apply to the original CREAM tables.

### 6.4.3 Technical implementation

We wanted the application to be multi-platform, portable, and stand-alone in the first iterations, while still being able to transpose it to a client-server model, or even to a desktop application if we later decide so. Consequently, we chose to implement *S-CREAM assistant* in JavaScript, storing all data in the web browser’s Local Storage. *S-CREAM assistant* uses several frameworks. AngularJS [80] manages the Views and the Controllers, js-data [109] handles the Models and provides



an Object-Relation Mapping, and D3.js [22] displays the interactive tree used to visualise S·CREAM's *Retrospective Analysis* step.

As shown in Figure 6.5, *S·CREAM assistant*'s data model uses different formats and is scattered across different storages. XML, XSLT, and JSON are used to configure *S·CREAM assistant* before running analyses, and Local Storage is used at runtime to store the work in progress. JSON is also used in the import/export feature and to store a serialised representation of the Java Script objects used at runtime (among these the interactive tree) in the Local Storage.

An XML file is used to store CREAM's original tables and XSLT files provide an easy way to the analyst to create S·CREAM *flavors*. An analyst can add a S·CREAM *flavor* by creating a corresponding XSLT style sheet, and thereafter registering the new style sheet in the main Java Script file of *S·CREAM assistant*, *app.js*.

This aforementioned file also stores the different types of descriptions (i.e., the STEAL-inspired one we use in this work) under the form of AngularJS constants that the analyst can modify at will before running *S·CREAM assistant*.

The interactive tree is stored in the Local Storage after each change to ensure no loss of work in case the analyst closes *S·CREAM assistant* or his web browser.

## 6.5 Conclusion

In this section, we have illustrated how we adapted CREAM, a technique used in safety to investigate the cause of 'human errors', to security. The resulting technique, named S·CREAM, has been bootstrapped and given a supporting tool, the *S·CREAM assistant*. In the next section, we use S·CREAM on several use-cases and identify the Contributors that can undermine the security of the systems these use-cases describe. Further, we assess S·CREAM's relevance for security.



*If everything seems under control, you're not going fast enough.*

—Mario Andretti

# 7

## Applying S·CREAM for socio-technical security analysis

### Contents

---

<b>7.1</b>	<b>Introduction</b>	<b>130</b>
<b>7.2</b>	<b>Authentication of identities with TLS certificates in web browsers</b>	<b>131</b>
7.2.1	Description	131
7.2.2	Threat Model	131
7.2.3	Semi-automatic Security Analysis	131
7.2.4	Analyst-driven Security Analysis	132
7.2.5	Discussion on the results and the possible remediations	132
<b>7.3</b>	<b>Hotspot use cases</b>	<b>133</b>
7.3.1	Description	133
7.3.2	Threat Model	134
7.3.3	Semi-automatic Security Analysis	135
7.3.4	Analyst-driven Security Analysis	135
7.3.5	Discussion on the results and the possible remediations	136
<b>7.4</b>	<b>YubiKeys use case</b>	<b>137</b>
7.4.1	Description	138
7.4.2	Threat Model	138
7.4.3	Semi-automatic Security Analysis	138
7.4.4	Analyst-driven Security Analysis	138
7.4.5	Discussion on the results and the possible remediations	139

**7.5 Conclusion . . . . . 140**

---

*In this chapter, we use S-CREAM on several use cases to stress out its relevance in security. The chosen use cases are the verification of identity via TLS certificates in web browsers, security ceremonies encountered when using Wi-Fi Hotspots, and YubiKey security tokens.*

## 7.1 Introduction

In Chapter 6, we presented S-CREAM which is our implementation of an Root Cause Analysis (RCA) for security that ultimately aims at answering the two following research questions stated in Chapter 2:  $RQ_1$ : *How can we detect a socio-technical vulnerability in a system?*, and  $RQ_3$ : *How can we identify the factors that foster human behaviours that are harmful to a system's security?*. In this Chapter, we use S-CREAM to illustrate how it proposes answers to these research questions by performing the *Security Analysis* of several use cases. The chosen use cases are the verification of identity via Transport Layer Security (TLS) certificates in web browsers (studied with STEAL in Chapter 3), security ceremonies encountered when using Wi-Fi Hotspots (studied with STEAL in Chapter 4), and YubiKey security tokens [188].

Studying these use cases calls for the definition of some methods. Indeed, S-CREAM, albeit ordering its steps to satisfy the steps' dependencies, does not impose a process to perform the socio-technical analysis of a system. For instance, an analyst can use a pre-existing catalog of Attack Modes (AMs) and perform a *Security Analysis* step to extend a threat model, or use the *Data Collection and Investigations* and the *Retrospective Analysis* steps to find the Contributors to an attack without using any of the other steps.

**Methods.** For each use case we, in turn, perform the semi-automatic (with the catalog of AMs bootstrapped in Chapter 6) and the analyst-driven *Security Analysis*. Given a threat model, the semi-automatic *Security Analysis* offers a rapid overview of the Contributors that, if found on the system under scrutiny, could be used by an attacker to gain the corresponding Socio-Technical Capabilities (STCs) in this system. The analyst-driven *Security Analysis* provides additional insights about the Contributors that could cause a potential attack on this system to succeed. As S-CREAM's analyst-driven *Security Analysis* does not provide a mean to identify or devise potential attacks, the analyst is left with the choice of the most appropriate method to find these attacks (or his own expertise). We choose to use STEAL to find potential attacks in these use cases, reusing the attacks already studied in the previous chapters and identifying new attacks for the new one: the Yubikeys use case. Thereafter, in a short security discussion, we explain how the identified Contributors could be controlled, and we discuss the overlap between the results obtained by using S-CREAM and the ones we obtained earlier using only STEAL (if applicable). Regarding the proposition of controls and remediations, we focus our efforts on Contributors found in the results of both the semi-automatic and

the analyst-driven *Security Analyses*. The rationale being that the Contributors identified by both the methods may be more likely to be exploited in the potential attacks, and since they are identified by the semi-automatic *Security Analysis*, we know that they were already exploited in the past attacks.

## 7.2 Authentication of identities with TLS certificates in web browsers

### 7.2.1 Description

We focus on one particular interaction between the user and the web browser that appears in the validation of identities of TLS certificates: the display of interstitial warnings (also discussed in Annex A). Some web browsers rely on these warnings to ask the user to take a decision about whether or not to continue to connect to a web server when the validation of the web server's certificate fails (see Chapter 3). This decision has security consequences that makes it critical, because connecting to the web server that presents an invalid certificate can harm data confidentiality and data integrity. Thus, the safe choice for a user facing such an interstitial warning when trying to connect to a web server is to abort the connection.

### 7.2.2 Threat Model

The main assumption for this system is that the attacker cannot modify the interactions between the user and the browser while the warning is displayed. What the attacker can do though, is to use the area where the web pages and the browser's interstitial warnings are displayed to send messages to the user prior to the warning interaction.

### 7.2.3 Semi-automatic Security Analysis

For compatibility with the catalogue of AMs we built in Section 6.3.5, we use the same description scheme. We consider that this system's Threat Model allows one to write messages on the web medium in which the attacker controls the source, the declared identity, the imitated identity, and the command. By assumption, the attacker cannot spoof the action the user is about to perform (the action triggered by the button displayed by the interstitial warning). We consider not having information about the sequence because the attacker can impersonate any source, and hence, we only have sequential information about the user-web browser interaction. In consequence, by using *S-CREAM assistant* with the bootstrapped catalogue of AMs populating with CAPEC, we learn that the reachable STC is ***Identity spoofing***. The Contributors identified by the semi-automatic *Security Analysis* are listed in Table 7.1.

Table 7.1: Lists of Contributors of the *Identity spoofing* STC and of the potential Man In The Middle attack. Shared Contributors are highlighted in blue.

STC: <i>Identity Spoofing</i>	Attack: Man In The Middle
GA-Faulty diagnosis	GA: Cognitive bias
GA-Inadequate quality control	GA: Faulty diagnosis
GA-Inattention	SA: Competing task
GA-Insufficient knowledge	SA: Confusing symptoms
GA-Mislabelling	SA: Erroneous analogy
GA-Missing information	SA: Erroneous information
GA-Wrong reasoning	SA: Error in mental model
SA-Ambiguous label	SA: Habit, expectancy
SA-Ambiguous signals	SA: Information overload
SA-Ambiguous symbol set	SA: Inadequate training
SA-Competing task	SA: Incorrect label
SA-Erroneous information	SA: Misleading symptoms
SA-Error in mental model	SA: Model error
SA-Habit, expectancy	SA: New situation
SA-Hidden information	SA: Overlook side consequent
SA-Inadequate training	SA: Presentation failure
SA-Incorrect label	
SA-Mislearning	
SA-Model error	
SA-Multiple signals	
SA-Overlook side consequent	
SA-Presentation failure	
SA-Too short planning horizon	

## 7.2.4 Analyst-driven Security Analysis

After the semi-automatic *Security Analysis*, we know that the attacker may be able to spoof an identity. We now investigate a potential attack (detailed in Annex A) that takes advantage of this capability and attempts to further exploit the user to escalate to a Man In The Middle attack. This potential attack, described in Annex A, targets the Google Chrome web browser. The attacker presents a fake interstitial warning to the user just before the genuine warning is displayed. The attacker’s fake interstitial warning appears no different from the genuine one to the user as the attacker controls both the declared and the imitated identity, and it intends to prompt the user to think that the word ‘self’ in ‘self-signed’ certificate means ‘a certification authority called SELF’ instead of its real English meaning. The consequence of this manipulation is that the user can misinterpret the genuine warning when it warns the user that a web server using ‘self-signed’ certificates should be avoided. The expected effect being that the user trusts the attacker’s self-signed certificate and allow the attacker to perform a Man In The Middle attack. The Contributors identified by the analyst-driven *Security Analysis* are listed in Table 7.1.

## 7.2.5 Discussion on the results and the possible remediations

Table 7.1 lists the Contributors obtained from the *Security Analyses*. As the Threat Model leaves the attacker with total freedom prior to the display of the genuine warning, there are a lot of Contributors that the attacker can exploit in its attempts to spoof an identity.

As the Man In The Middle attack shares several Contributors with the *Identity spoofing* STC, we assume that controlling these Contributors is likely to weaken

the chances of this attack to be successful along with all the others attacks based on this STC. We group these Contributors into the following categories and then discuss potential remediations (a Contributor can belong to several categories):

- Attacker’s capabilities: ‘Presentation failure’, ‘Erroneous information’, and ‘Incorrect label’ are related to the technical capability of the attacker to present fake information as being genuine to the user. One immediate remedy would be to change the way warnings are displayed to make their impersonation more difficult or impossible. They should be displayed in a place where the attacker can’t write. If this is not possible, adding additional signs outside the display area that a warning is genuine can help. The Microsoft Windows User Account Control display is an example of this strategy.
- Human features: ‘Habit, Expectancy’, and ‘Overlook side consequent’ are related to the tendency of people to go with the flow and are an example of what we can’t control. ‘Inadequate training’ could be solved by better education.
- System’s features: ‘Inadequate training’, and ‘Error in mental model’ can be related to the system. The use of a web browser should not need any specific training. The web browser, its use of TLS as well as its behaviours and interactions should be simple and predictable and provide a good user experience. In a nutshell, it should be usable. Concepts that end up being manipulated by the end users should be well-defined, and they should be introduced only if needed. See for instance, Jackson *et al.*’s work on the matter [102].
- Organisational: ‘Inadequate training’ is related to the lack of training the user might have prior to this encounter, which could help him avoiding the threat.

## 7.3 Hotspot use cases

This work is based on the work presented in Chapter 4, where we identified several ‘critical actions’ in four use cases of Wi-Fi Hotspots. In the previous works, we formulated hypotheses about the factors that could lead to successful attacks, but could not investigate all of these factors through user-studies or surveys. Only the link between network names, context, and trust in the network (see Chapters 3, and 4) as well as the influence of graphical cues during the selection process (see Chapter 4) were studied.

### 7.3.1 Description

A detailed description of the four use cases can be found in Chapter 4. Here, we only provide the gist of each situation:

- **HotSpot 1.1: Pay-per-use Hotspot** The user selects a network to connect to, performs a payment through HTTPS, and then uses the unencrypted network.

Table 7.2: Potential factors that we identified in the previous works that can potentially foster the insecure behaviours.

Critical Action	potential factors
Changing settings	HS 2.1: Money, free services
Selecting a malicious Access Point	HS 1.1: Context, likeliness of the SSID name, signal strength HS 2.2: Images, venue name, roaming partners, quantity of information
Using a malicious Access Point	HS 1.1: Encrypted money transaction, encryption, encryption cues HS 1.2: Trust in the ISP, trust in authority, encrypted interactions HS 2.1: Complexity, money, network, surprise, automatic connection

- HotSpot 1.2: Internet Service Provider’s Homespots** The user is a customer of an ISP that gives access to Wi-Fi hotspots hosted by the other ISP’s customers. The user selects the network named as instructed by the user’s ISP, and is then redirected to a website where he enters his credentials through an HTTPS session. The user eventually uses the unencrypted network.
- HotSpot 2.1: Mobile Network Operator’s partner Hotspot 2.0 [37]** This use case is related to the particular feature of some smart phones that can steer traffic off the 3G and 4G networks to nearby Wi-Fi networks following a routing policy, the ANDSF<sup>1</sup> policy [1]. The connection to the Wi-Fi network is done without any user intervention to a secure network. However, the user can apply preferences in terms of the policy to prefer personal networks, or certain types of networks (unencrypted for instance).
- HotSpot 2.2: The future of Hotspots** This use case focuses on the cohabitation between conventional Hotspots with Hotspots 2.0, which support a feature called the Mobility Services Advertisement Protocol. This feature allows network providers to advertise the networks’ services through a name and an icon, and then to redirect users that connect to the associated network to an url.

We are interested in the three following critical actions that a user can perform: *changing phone settings for a loose ANSDF policy* (HS2.1), *selecting a malicious Access Point* (HS1.1, HS2.2), and *using a malicious Access Point* (HS 1.1, HS 1.2, and HS 2.1). Table 7.2 lists the factors that we identified in our previous works as being the likely influencers of the user’s decision to perform these critical actions.

### 7.3.2 Threat Model

We consider that the attacker can read and write in the ether and on the Internet, but it can’t break the cryptography. It means that it can, for instance, bring up access points, read unencrypted Wi-Fi traffic, and post urls on Internet forum boards; however, it cannot read HTTPS traffic or impersonate an Access Point that is authenticated by cryptographic means.

<sup>1</sup>Access Network Discovery and Selection Function



Table 7.3: Results of each analyst-driven *Security Analysis* for each Hotspot use case grouped by insecure behaviours.

Changing settings HS 2.1	Selecting a malicious AP	
	HS 1.1	HS 2.2
GA: Insufficient knowledge	SA: Ambiguous signals	GA: Distraction
SA: Overlook side consequent	SA: Ambiguous symbol set	GA: Insufficient knowledge
SA: Too short planning horizon	SA: Erroneous information	SA: Ambiguous signals
SA: Violation	SA: Habit, expectancy	SA: Ambiguous symbol set
	SA: Incorrect label	SA: Erroneous information
	SA: Overlook side consequent	SA: Error in goal
		SA: Habit, expectancy
		SA: Incorrect label
		SA: Information overload
		SA: Overlook side consequent
Using a malicious AP		
HS 1.1	HS 1.2	HS 2.1
GA: Cognitive bias	GA: Cognitive bias	GA: Cognitive bias
SA: Ambiguous signals	SA: Ambiguous signals	SA: Competing task
SA: Ambiguous symbol set	SA: Ambiguous symbol set	SA: Habit, expectancy
SA: Competing task	SA: Competing task	SA: Hidden information
SA: Error in mental model	SA: Error in mental model	SA: Multiple signals
SA: Habit, expectancy	SA: Habit, expectancy	SA: Overlook side consequent
SA: Hidden information	SA: Hidden information	SA: Presentation failure
SA: Multiple signals	SA: Multiple signals	
SA: Overlook side consequent	SA: Overlook side consequent	
SA: Parallel tasks	SA: Parallel tasks	
SA: Too short planning horizon	SA: Too short planning horizon	

### 7.3.3 Semi-automatic Security Analysis

To build the extended Threat Model, we settle on the lowest common denominator between the different investigated critical actions. We consider that the attacker has control over the source, the declared identity, and that he can write in the ether. However, the attacker cannot spoof actions or initiate interactions. In consequence, the reachable STC is *Identity spoofing*. Once we describe this system into the *S-CREAM assistant*, we learn that the attacker can manipulate the following Contributors to gain the *Identity spoofing* STC: SA-Incorrect label, GA-Missing information, SA-Erroneous information, and SA-Presentation failure.

### 7.3.4 Analyst-driven Security Analysis

Each insecure behaviour (for the applicable use cases) is contextualised in a potential attack where the attacker sets up malicious Access Point and listen to unencrypted traffic. Table 7.3 shows the results of each analyst-driven *Security Analysis* conducted with S-CREAM grouped by insecure behaviours.

## 7.3.5 Discussion on the results and the possible remediations

### 7.3.5.1 Contributors to the STC

The semi-automatic *Security Analysis* does not bring interesting results when considering traditional Hotspots as it only states the obvious: since the attacker can provide anything as network name, he can spoof identities of other principals. Even when we consider Hotspot 2.0, the conclusions are still the same, i.e., the attacker can still write any network name and spoof identities. But contrary to traditional hotspots, Hotspot 2.0 can provide mutual authentication to prevent the user from joining a malicious Access Point. Nonetheless, if for any reason, this control fails (for instance, if the attacker downgrades the authentication mechanism to a mechanism without mutual authentication), it is worth investigating further the Contributors that can foster the exploitation of the STC in such a case.

### 7.3.5.2 Changing settings

The results that S-CREAM yields for this behaviour are related to the cognitive process that drives the user's decisions. The user can modify these settings with a goal in mind and without understanding the side effects the decision can have in the future. These settings could be redesigned to offer the user more information about these aspects of his decision.

### 7.3.5.3 Selecting a malicious Access Point

The Contributors for the selection of a malicious Access Point are mostly linked to technological factors. S-CREAM identifies not only the obvious 'erroneous information' Contributors but also the 'ambiguous symbol set' that we know, for having tested it in Chapter 4, to have the capability to influence user's decision when selecting a network. Among the Contributors that we can attribute to the user, only the knowledge-related Contributors could be addressed, in particular by user training.

Comparing the results between HS1.1 and HS2.2, we observe at first glance that there are almost twice as much Contributors for this critical action in HS 2.2 than there are in HS 1.1. This shows that the new features introduced by Hotspot 2.0 expand the Socio-Technical Attack (STA) surface. Hotspot 2.0 may provide technical means to authenticate the AP, and under the perfect encryption assumption, is a better option for building Hotspots than the old-fashioned Hotspots. However, as far as the social part of the interaction is concerned, the attacker can use the new features to trick people into choosing his network over the others.

### 7.3.5.4 Using a malicious Access Point

The main novelty when considering the use of a malicious Access Point is the interference from a competing task for which we can't propose remediations.

### 7.3.5.5 Comparison with previous works

As explained earlier, we had expectations regarding the possible factors that could nudge a user to jeopardise his security in the different use cases. Unfortunately, as we did not have enough resources to test every hypotheses we had produced with STEAL (see hypotheses listed in Tables 7.2), we can only compare results yielded with S·CREAM to a few verified results, and from them draw, comments on the nature of the others.

For instance, we expected money to be a motivator for the user to change its mobile phone's settings for a loose ANDSF policy because driving traffic off the ISP network would save the user some money from his monthly phone bill. And instead of 'money', S·CREAM points out four Contributors focused on the cognitive process that drives user's decisions. This hints towards a difference between the Contributors that we can expect from S·CREAM in its present state, which states that S·CREAM won't point out Contributors linked to user's motivation. Indeed, humans do not act erroneously on purpose. The motivated erroneous behaviours fall into the 'violation' category, and these behaviours are not investigated further by the underlying CREAM methodology.

## 7.4 YubiKeys use case

A YubiKey is a multi-purpose security token in the form of a USB dongle. A YubiKey is versatile as it can present itself as a keyboard or a two-factor authentication device to a computer (or via NFC to a smart phone). A YubiKey can be used to generate and store a 64 characters password, generate One Time Password, or to play different challenge-response protocols [190].

The main advantages of YubiKeys are their ease of use, low price, and wide adoption by big service providers. Indeed, a YubiKey's user interface consists of only one button and a LED. Yubikey's solid construction allows to carry a YubiKey all the time, and it is supported by services like Google, Dropbox, or Github. The YubiKey is advertised as a push button solution for security.

We are interested in YubiKeys mainly because of their peculiar user interface and user interactions. Yubico made the choice to not include a screen on the device to lower its cost and foster its adoption. The absence of a screen has for consequence to shift the duty of providing feedback to the user to a LED light. The LED's behaviors (e.g., flashing rapidly, being on or off, *et cetera*) have different meanings and are explained in the user's manual [190].

One aspect of Yubikeys is that they support two configuration slots on one device. To use these configurations, the user touches the button of the device for different periods of time. These slots can be configured to generate OTPs or a static password. See below a quote from yubico's YubiKeys security evaluation document [189] that describes this functionality as well as its assumed security implications:

The YubiKey 2.0 introduces a mechanism where the user can use two separate credentials. We call the storage for these credentials 'slot 1' and 'slot 2'. To generate a credential from slot 1, the user touches the

button for a short period of time (e.g., well below 2 seconds). To generate a credential from slot 2, the user touches the button for a long period of time (e.g., well above 3 seconds). With proper user education, we believe this does not add any additional security problems so we continue to evaluate the YubiKey configured with just the slot 1 credential.

It is worth noting that YubiKeys (now in version 4) come in two forms: the standard YubiKey is 18 x 45 x 3 mm, has a round-shaped button, and its LED is located at the top, and the YubiKey ‘nano’ is much smaller, completely disappears into a USB port when plugged in, and it has its button and its LED on its edge.

### 7.4.1 Description

As YubiKeys have a lot of functionalities and different use cases, we focus our analysis on the basic operation of a YubiKey with the ‘Dual configuration’ functionality enabled. We set a YubiKey nano to yield an OTP on slot 1, and a static password on slot 2.

### 7.4.2 Threat Model

The main assumptions for this system are that the attacker can read and write on the Internet. This Threat Model implies that the attacker is free to send messages on the web medium to the user before and after the operation of the Yubikey by touching its button. More specifically, we consider that the user is visiting a website under the control of the attacker.

### 7.4.3 Semi-automatic Security Analysis

We consider that this system’s Threat Model allows the attacker to control the source, the declared identity, the imitated identity, the command, and that it can write on the web medium. As the attacker has no control over the YubiKey, he cannot spoof the action the user is about to perform. The attacker has control of the sequence of communication with the user. In consequence, by using S-CREAM, we find that the reachable STC is *Identity spoofing*.

### 7.4.4 Analyst-driven Security Analysis

To find likely potential attacks on this system, our strategy is to formulate hypotheses about the consequences of the user’s actions in consideration of the attacker’s extended capabilities.

There are two actions that a user has to carry out when using a YubiKey on a computer: **plugging** the YubiKey into a USB port, and **operating** the YubiKey by touching its button according to the authentication scheme. On the YubiKey nano, both actions are critical from a security point of view.

- **plugging** the Yubikey nano in a computer can accidentally produce an OTP because of the location of the button at the edge of the device. **Plugging** or

Table 7.4: Contributors yielded by the *Security Analysis*.

STC: <i>Identity spoofing</i>	Attack: Foster ‘Sequence-Wrong action’ EM on <b>plugging</b>	Attack: Foster ‘Duration-Too long’ EM on <b>operating</b>
GA-Faulty diagnosis	GA: Sound	GA: Adverse ambient conditions
GA-Inadequate quality control	SA: Competing task	SA: Confusing symptoms
GA-Inattention	SA: Design	SA: Inadequate training
GA-Insufficient knowledge	SA: Noise	SA: Information overload
GA-Mislabelling		SA: Mislearning
GA-Missing information		SA: Multiple signals
GA-Wrong reasoning		SA: New situation
SA-Ambiguous label		SA: Noise
SA-Ambiguous signals		SA: Overlook side consequent
SA-Ambiguous symbol set		SA: Too short planning horizon
SA-Competing task		SA: Trapping error
SA-Erroneous information		
SA-Error in mental model		
SA-Habit, expectancy		
SA-Hidden information		
SA-Inadequate training		
SA-Incorrect label		
SA-Mislearning		
SA-Model error		
SA-Overlook side consequent		
SA-Presentation failure		
SA-Too short planning horizon		

**unplugging** a YubiKey nano can lead to a loss of confidentiality of the OTP code located in the first slot. As the YubiKey operates after the touching event is finished we consider that the Error Mode to investigate is ‘Sequence-Wrong action’, and that the user appends an irrelevant action to the sequence of actions.

- **operating** the YubiKey nano has two important dimensions: the action’s duration (i.e., less than 2 seconds or more than 3 seconds) and the action’s location (i.e., which user interface element has the focus at the time of the action). The user needs to touch the device for the right amount of time while being in communication with the correct entity; otherwise, there can be a loss of confidentiality. As location-based attacks are already covered by the *Identity spoofing* (i.e., the user misidentifies the attacker for another entity), we focus on the duration. In particular, we investigate the Error Mode (EM) ‘Duration-Too long’.

We sum up the results of this investigation in Table 7.4.

### 7.4.5 Discussion on the results and the possible remediations

Regarding *Identity spoofing*, the attacker has a lot of options when it comes to impersonating another entity (see the *Identity spoofing*’s column in Table 7.4). A prominent example of such attack is the Man In the Browser attack: the attacker, in control of the web browser, redirects the user to a website he controls when the user attempts to go to his bank’s website. The attacker then asks for the credentials

(including two-factors Authentication credentials as the one provided by a YubiKey) and logs into the bank’s website in place of the user. The key result of this analysis is that there is little that can be done to thwart the attack, given the number of Contributors. Therefore, we can either accept the risk that poses this vulnerability or design alternatives to two-factors authentication mechanisms where the attacker can not socio-technically impersonate other entities. The results of this analysis come to the same conclusion as the security evaluation made by yubico [189], which states, ‘We conclude that the system does not provide good defence against a real-time man-in-the-middle or phishing attack.’

The observation of the failure of 2 Factors Authentication mechanisms leads to the inception of Fido U2F [158][191] that allows the security device (such as a YubiKey) to authenticate the website before running any authentication schemes.

Regarding potential attacks on the ‘Dual configuration’ functionality, Table 7.4 shows that there are three Contributors that an attacker can manipulate to foster the occurrence of the ‘Sequence-Wrong action’ EM during the **plugging** critical action. The attacker, in control of the webpage can emit sounds or noises to apply pressure on the user, and he can also create a competing task. While we can see little practical application of this attack, we consider that these AMs are exploitable and it’s yubico’s duty to consider whether or not this socio-technical vulnerability should be controlled.

Finally, we turn to the case of the **operating** critical action. Investigating this *critical action* with S-CREAM yielded more Contributors than the **plugging** critical action, and therefore, it appears more likely to observe potential attacks that exploit the **operating** action as opposed to the **plugging** action. Table 7.4 lists the Contributors that we reckon can be used to trigger to the ‘Duration-Too long’ EM. For instance, we select ‘SA-Confusing symptoms’ because the attacker can provide fake feedback to the user regarding the success of the fake authentication, which could in turn lead the user to start a remediation loop in which he would try everything in his power to authenticate successfully; in the process, leaking every confidential information the YubiKey’s slots hold. For example, sometimes users try every password they know when an attacker sends a ‘bad authentication’ message as the sole feedback. This kind of an attack is very well possible given the fact that the YubiKey provides little feedback when a slot is yielded and no feedback about which slot is yielded. Furthermore, the user might be unsure how he configured his YubiKey (and someone may have configured it for him).

## 7.5 Conclusion

We have illustrated how to use S-CREAM to perform a socio-technical Security Analysis of different systems. We chose a method that draws from S-CREAM’s accumulation of knowledge from previous attacks and from the analyst’s expertise. S-CREAM offered interesting insights on the use cases already studied with STEAL and novel insights on the security of Yubikey security tokens.

The S-CREAM analysis of YubiKeys is very important for validating the S-CREAM methodology because it neatly aligns design and usability choices with security consequences. It sheds a new light on the possible vulnerabilities, espe-

cially socio-technical vulnerabilities, which can affect these devices beyond simple Phishing attacks, and it shows that Yubikeys, despite being technically secure, fail to provide *effective security* because of functionality as well as usability choices.

Furthermore, the Yubikey use case showed how S·CREAM can be used to analyse the security of tangible objects that offer haptic interactions, and how S·CREAM can give new insights on the effects of these modalities of interactions on the system's security. Finally, the Yubikey use case is a great example of how S·CREAM can be used to avoid the analyst's bias. Indeed, contrary to the security analyst's gut feeling that the Yubikey's 'Dual configuration' feature does not have security implication [189], S·CREAM *Security Analysis* results show that there are a lot of potential ways through which an attacker could exploit this feature.

In the next Chapter, we discuss how S·CREAM meets the challenges we defined for RCA to be a valuable tool for security.





*I always said that the data tables in THERP were not written in stone, and I was not Moses coming down from a mountain with these tables so inscribed.*

—Alan Swain, creator of THERP [166]

# 8

## Review of how S·CREAM meets our challenges

### Contents

---

<b>8.1</b>	<b>Introduction</b>	<b>143</b>
<b>8.2</b>	<b>How S·CREAM meets our challenges</b>	<b>144</b>
8.2.1	Challenge $C_4$ : Matching patterns of known attacks	144
8.2.2	Challenge $C_3$ : Creating reusable knowledges	146
8.2.3	Challenge $C_2$ : Investigating Attacks	147
8.2.4	Challenge $C_1$ : Addressing the lack of knowledge and structured data to support the analysis	149
8.2.5	Challenge $C_5$ : Being flexible	150
<b>8.3</b>	<b>Conclusion</b>	<b>150</b>

---

*In this chapter, we discuss whether S·CREAM meets the challenges for a Root Cause Analysis for security that we defined in Chapter 5. We discuss each of the four steps that S·CREAM comprises, starting from S·CREAM's ultimate step, the Security Analysis, following reverse order. We discuss each step's shortcomings and potential improvements as well as identify the inherited shortcomings, to be discussed upon reaching the step at their source down the stack of S·CREAM's steps.*

### 8.1 Introduction

In this chapter, we discuss how S·CREAM meets the challenges to build a Root Cause Analysis (RCA) for security that we identified in Chapter 5. As stated in

Chapter 6, S-CREAM assigns the duty to address each one of the four first challenges to an operational step, and  $C_5$ , being flexible, is addressed by the whole methodology and its implementation. Table 8.1 is an overview of the links between challenges and S-CREAM's steps.

Table 8.1: Links between the challenges stated in Chapter 5 and S-CREAM's steps introduced in Chapter 6. Each challenge depends on its predecessor except  $C_5$ , which is the subsidiary challenge of being flexible.

Challenge	Challenge description	S-CREAM step
$C_1$	Addressing the lack of knowledge and structured data	<i>Data Collection and Investigations</i>
$C_2$	Investigating Attacks	<i>Retrospective Analysis</i>
$C_3$	Creating reusable knowledge	<i>Generalisation</i>
$C_4$	Match patterns of known attacks	<i>Security Analysis</i>
$C_5$	Being flexible	All steps

The challenge that corresponds to S-CREAM's ultimate step *Security Analysis*, is to be able to match patterns of known attacks or identify socio-technical vulnerabilities in a system. As this step is the last in a chain of backward-dependent steps, its shortcomings may be caused by implementation choices made at lower level in the stack of steps. For instance, implementation choices made in the *Data Collection and Investigations* step. For this reason, in the following, we choose to discuss steps' shortcomings and their causes in the backward order, carrying inherited deficiencies down the stack until the discussion focuses on the concerned step.

## 8.2 How S-CREAM meets our challenges

### 8.2.1 Challenge $C_4$ : Matching patterns of known attacks

#### 8.2.1.1 Evaluation of S-CREAM's fulfilment of the challenge

Challenge  $C_4$  states that *the RCA technique should provide direct links between the attacker's capabilities and their effects on a system's security*. We consider that S-CREAM addresses the challenge as we demonstrate in Chapter 7 how to apply S-CREAM on several use cases, and how a security analyst can, from the different threat models, pinpoint potential socio-technical vulnerabilities.

But there is one limitation in the methodology that we need to address, which is its partial reliance on the analyst's skills and expertise. Indeed, in the study of Chapter 7's use cases, the most interesting findings were the ones yielded through the use of the analyst-driven *Security Analysis*, while the semi-automatic *Security Analyses* yielded the ***Identity spoofing*** for all use cases. Furthermore, once the semi-automatic *Security Analysis* lists the potential Contributors for a system, the analyst is left alone with the duty of pondering on whether or not these Contributors can actually constitute socio-technical vulnerabilities.

Moreover, one can wonder whether the choice of limiting the pre-requisites presented in the Attack Modes (AMs) to the attacker’s capabilities was the right thing to do.

Eventually, there is the problem of the complexity of the Contributors’s wording and concepts. Indeed, S-CREAM is based on Cognitive Reliability and Error Analysis Method (CREAM)’s tables, and consequently, S-CREAM’s Contributors use CREAM’s antecedents, which can sometimes be daunting. For instance, CREAM makes a difference between ‘mis learning’ (the user is at fault), and ‘inadequate training’ (the training is at fault). In this case, one antecedent hints that there is a possible remediation (improve the training), however, if while performing the *Retrospective Analysis* of an attack, the analyst chooses ‘mis learning’ over ‘inadequate training’, then no remediation will be possible in the *Security Analysis* when the corresponding AM is identified in a system.

### 8.2.1.2 Inherited shortcomings and potential improvements

S-CREAM and its companion tool, *S-CREAM assistant*, only enables the analyst to define pre-requisites for the attacker’s capabilities. Broadening the scope of these pre-requisites by allowing the analyst to define pre-requisites pertaining to the context, the user and the system itself would improve the resolution of the AMs. This improved precision would make the semi-automatic *Security Analysis* step more objective because with better security analysis, the results would be stripped of false-positives and the analyst would be relieved from the burden to filter out useless Contributors.

Let us focus on the example of the ‘Competing task’ Contributor. If the ‘Competing task’ Contributor is identified as a Contributor of an Socio-Technical Capability (STC) in a system, the analyst has to decide whether the system should allow for tasks to be run concurrently or not, or if the attacker can create competing tasks. If we modify the pre-requisites’ structure to allow the analyst to define pre-requisites about the system’s stance regarding concurrent tasks, the analyst is relieved from this task and the analysis gains more objectivity.

Another aspect of the difficulty to assess the criticality of a Contributor is that it can be distant to the STC it is linked to. Indeed, as an attack can rely on the exploitation of several Error Modes to succeed, trying to build remediations from a merged list of Contributors may not be the best solution to assess the vulnerability of a system. S-CREAM could be altered to replace the merged list of AMs by sets of AMs and create an object that holds the relationships between these. That is to say, this object could state that an attacker needs to exploit an AM of each set to gain the STC. Controlling all the Contributors of a set of AMs would then disrupt the potential gain of the corresponding STC by preventing one of the mandatory Error Modes from occurring.

Finally, regarding S-CREAM’s Contributors, we believe that each should be assessed thoroughly to determine their utility in a *Security Analysis*. Taking back the example of the ‘mis learning’ Contributor, we believe that it would be beneficial if it was removed from the tables.

### 8.2.1.3 Step's shortcomings and potential improvements

All analysts are not equal, and determining if a Contributor, for instance, related to usability, constitutes a socio-technical vulnerability is not a straightforward task. Without modifying the pre-requisites, the *Security Analysis* could provide more guidance to the analyst while assessing the relevance of Contributors in a system's security. We identified two major improvements that could be made to *S-CREAM assistant* that would help the analyst in this task. First, *S-CREAM assistant* could provide a helper that displays, for instance, a checklist for each Contributor. Second, *S-CREAM assistant* could allow experts in the field related to a potential Contributor to cooperate with the analyst in determining the likelihood that this Contributor influenced the outcome of the attack and to complement the analysis with additional input.

## 8.2.2 Challenge $C_3$ : Creating reusable knowledges

### 8.2.2.1 Evaluation of S-CREAM's fulfilment of the challenge

As shown in Chapter 7, we successfully used the catalogue of AMs bootstrapped in Chapter 6 to extend the threat models associated to different use cases. Therefore, we consider that S-CREAM's *Generalisation* step meets Challenge  $C_3$ .

Nonetheless, one can argue that the generated knowledge is limited because there are only two STCs, and they are very generic. Furthermore, we did not devise the expected STCs mentioned in Chapter 6, namely *Block* and *Alter*.

One can also wonder what the difference is between the 'Action Spoofing' field from the description of an attack and the *Action spoofing* STC. The answer is that they are actually different dimensions of the same malicious behaviour. As a description, 'Action Spoofing' describes the **technical capability** of the attacker to change the target of an action, whereas as a **Socio-Technical Capability**, it represents the capability of decoying the user into performing this booby-trapped action. These two dimensions are needed for an attack based on a spoofed action to work. Where one could control technical factors to prevent the attacker from spoofing an action, S-CREAM's results offer additional socio-technical factors that one can use for the same purpose.

### 8.2.2.2 Inherited shortcomings and potential improvements

We have not identified any inherited shortcoming.

### 8.2.2.3 Step's shortcomings and potential improvements

There are two decisions that impacted the STCs provided by S-CREAM. The first one is that we focused on STCs pertaining to the attacker's effect on the user, however we can specialise the STCs further. In particular, we believe that it would be valuable to have STCs specialised by environments. For instance, as the user-device interactions differ from mobile phones to desktops, having STCs dedicated for each environment would better describe the attacker's capabilities in terms of actions on the system.

The lack of the expected STCs **Block** and **Alter** in the catalogue of AMs is the result of the decision of using CAPEC as the corpus of attack from which to build the catalogue. Now, there are works, in particular Heartfield *et al.*'s work on semantic attacks [86], that would constitute a more diverse corpus to build the catalogue. Indeed, CAPEC is lacking a lot of interesting Socio-Technical Attacks (STAs), such as tabnabbing attacks [146], fake loading bars, fake ads, *et cetera*.

Further, S-CREAM has still not been used to investigate real attacks, and we think that this is a mandatory step for it to get a better, more useful catalogue of AMs.

### 8.2.3 Challenge $C_2$ : Investigating Attacks

#### 8.2.3.1 Evaluation of S-CREAM's fulfilment of the challenge

As demonstrated in Chapter 6 with the bootstrapping of the catalogue of AMs, and in Chapter 7 with the analyst-driven *Security Analyses* performed on several use cases, S-CREAM's *Retrospective Analysis* fulfils Challenge  $C_2$ . Indeed, S-CREAM's *Retrospective Analysis* yields *human-related factors that are likely to explain the success of attacks, or potential attacks*.

Nonetheless, one can argue that it fails to be objective as the analyst plays an important role in the *Retrospective Analysis* process. Indeed, while performing the *Retrospective Analysis*, the analyst has to choose first the Error Modes (EMs) observable in the attack, then the antecedents that contribute to the attack's success. As mentioned earlier, the analyst can misunderstand the antecedent and miss it or mistakenly add it as Contributor. And the same can happen while selecting EMs. All these errors can potentially have harmful consequences on subsequent *Security Analyses*.

Further, one can argue about the relevance of CREAM's antecedents and that of EMs for computer security.

#### 8.2.3.2 Inherited shortcomings and potential improvements

As already mentioned when discussing  $C_4$ , broadening the scope of the attack's description with information about the context, the user, and the system could help addressing the question of objectivity. Indeed, with this additional bit of information, S-CREAM could be adapted to have links between Contributors and item descriptions. For instance, knowing that the victim performed additional tasks during an attack, S-CREAM could pre-select the 'Competing task' Contributor (if applicable to the error modes under scrutiny). S-CREAM would then provide additional guidance to the analyst, thus improving the method's objectivity and the quality of the results.

#### 8.2.3.3 Step's shortcomings and potential improvements

Regarding S-CREAM objectivity, we think that the remedies proposed for the *Security Analysis* also apply to the *Retrospective Analysis*: *S-CREAM assistant* should allow the analyst to ask for the cooperation of experts or to compare his analysis results with results of others to enhance the quality of the results. Further,

*S-CREAM assistant* should provide helpers for each antecedent to help the analyst in determining whether or not the antecedent is a Contributor of the attack. For instance, to select the ‘error in mental model’ antecedent, the analyst should have access to a tool to help determine how complex a system is for its users. Or at minimum, S-CREAM should provide some documentation or bibliographic references to improve the analyst’s understanding of an antecedent.

Regarding S-CREAM’s EMs, we can observe that the catalogue of AMs presented in Section 6.3.5 is based on the investigations of 15 Attack Patterns, and yet, we only used EMs belonging to two categories out of eight to build it. These two categories are ‘Sequence’ and ‘Wrong Object’. Further, one Error Mode, ‘Wrong object-Similar object’ is present in almost all situations.

This shows that, from CREAM’s EMs point of view, the STAs we investigated are similar to one another. But CAPEC Attack Patterns that encompass a user are rudimentary, and again, one can wonder if our results would be different using a different source as the corpus of attacks. Indeed, exploitation of psychological characteristics in STAs are on the rise (e.g., tabnabbing attack), but are nowhere to be found in CAPEC.

Furthermore, it’s only when it comes to the yubikey use case in Chapter 7 that additional EMs are used (‘Duration-Too long’ and ‘Sequence-Wrong action’), showing that these EMs could indeed be useful for computer security. We believe that, even if not all are proven useful today, all EMs present in the Cognitive Reliability and Error Analysis Method for Socio-Technical security (S-CREAM) should be kept to deal with future interactions. Indeed, two-factors authentication, Biometrics, Virtual Reality, Augmented Reality, and touch interactions with haptic feedback are about to become mainstream, and these additional EMs could be useful to investigate future attacks performed through their manipulation.

Consequently, we believe that S-CREAM’s Error Modes, inherited from CREAM, are sufficient for studying user-mediated attacks, and that even if some EMs do not prove useful today, all of them should be kept.

But the previous observation may not be true for Contributors. As we saw on the hotspot use case in Chapter 7, we did not find any Contributors related to money, or Contributors aimed at explaining user’s motivation when we expected to find some (alas we only hypothesised it).

In CREAM, antecedents related to the user are found in the “Personal related genotype”. Hollnagel explains [91] that regarding the main cognitive functions (observation, planning, and interpretation), CREAM mostly contains *Specific Antecedents* because:

This is an area where there has been a considerable amount of research, and where several specific models or theories have been proposed [...]. From the point of view of CREAM, practically all information processing theories and models propose specific antecedents only, since they describe relatively shallow hierarchical category structures.

As a consequence, several antecedents that we would expect to be generic are specific. For instance, violations are considered specific, which stops further investigation. We believe that it is also because in safety, there is no actor trying

to motivate an operator to fail or behave erroneously and a money-driven violation could actually be punished by the law. There is then, no need of money in CREAM's tables. We argue that with the extensive set of research on scams, insider threat, and violation of security policies that we introduced in Chapter 2, S-CREAM should be further improved, and antecedents specialised in computer security should be added. It is worth noting that, at the moment, the *Data Collection and Investigations* step do not offer the possibility to describe the transfer of value between an attacker and a user, and that this could be an interesting addition.

## 8.2.4 Challenge $C_1$ : Addressing the lack of knowledge and structured data to support the analysis

### 8.2.4.1 Evaluation of S-CREAM's fulfilment of the challenge

We are aware that S-CREAM does not offer the detailed collection of information about STAs that would be needed to exploit the technique to its full potential. This lack for instance undermines the potential creation of *Indicators of STA*. In the current scheme, STAs are described in terms of the attacker's actions and attack's effects on the system. Further, this description is flattened as it does not allow a description of a succession of actions as it would be possible, for instance, with attack trees. As mentioned when discussing  $C_4$  and  $C_2$ , this limitation of the *Data Collection and Investigations* propagates on all the steps of the technique. Consequently, we performed our S-CREAM analyses on generic descriptions of attacks and potential attacks (furthermore CAPEC is also very generic in its description of its Attack Patterns (APs)), while RCA techniques like CREAM are designed to be used on real events with a lot of details.

Keeping the description of attack simple is a limitation when it comes to using the technique on real attacks, however, it was 'good enough' to prove the usefulness of RCA techniques in security. The main rationale behind this choice is that we wanted to keep the technique simple for a first run, and that we believe that offering detailed descriptions of attacks is a long-term goal for S-CREAM. As stated in  $C_1$ , the RCA should provide a list of factual information to collect in order to maximise its outputs. This information can be related to the user, the environment, the system, and the attacker, but at the moment, there is no existing method to collect evidence and testimonies from users after STAs. Additionally, the scheme in which to describe these STA is to be defined.

### 8.2.4.2 Inherited shortcomings and potential improvements

We have not identified any inherited shortcoming.

### 8.2.4.3 Step's shortcomings and their potential improvements

First, instead of only describing the messages exchanged between the attacker and its victim, we should use in the *Data Collection and Investigations* step, a scheme that allows to add a description of the system, the context, and the user.

Further, a method to collect this information, including retrieving information from and about the victim, has to be defined. This is an additional sub-challenge S-CREAM should solve and, we chose to keep this as a direction for future works.

Additionally, regarding the support of *Indicator of STA*, we think that more than the *Data Collection and Investigations* step, it is the *Generalisation* step that needs improvements. In this regard, we identified two shortcomings in particular. First, Contributors are difficult to turn into *Indicators of STA* because it is already difficult to assist humans in the task of identifying them in a system. Indeed, documentation to precisely define Contributors and helpers for locating them are still lacking. Second, we believe that AMs still need to be improved before turning their identified Contributors into *Indicators of STA*. Indeed, we believe that AMs would be more useful and would help avoiding false positives in the detection of STAs, if instead of yielding only one Contributor, AMs yielded patterns or sets of Contributors for an STC.

## 8.2.5 Challenge $C_5$ : Being flexible

### 8.2.5.1 Evaluation of S-CREAM's fulfilment of the challenge

Thanks to *S-CREAM assistant's* design and the way the catalogue of AMs is constituted, most of the aforementioned improvements are easily implementable. Indeed, the need to change *Data Collection and Investigations* step's scheme along with the need to improve the catalogue of AMs or specialise S-CREAM's table are things that were foreseen from the beginning, and are easily customisable in *S-CREAM assistant*.

However, the implementation of helpers to guide this analyst in his decision regarding the contribution of an antecedent to an attack or regarding the presence of a Contributor in a system will call for more investments. The same is true for the need to cross-check analysis between analysts, share analyses among peers, and ask for help to an expert of a field of research associated with an antecedent.

*S-CREAM assistant* will need further developments and S-CREAM will need to be used on a wide range of attacks. Additionally, its catalogue will need to be cleverly enriched before it entirely proves its usefulness in the analysis of the security of socio-technical systems.

## 8.3 Conclusion

By examining each of the four steps in S-CREAM, we showed that S-CREAM meets the challenges stated in Chapter 5. Nonetheless, several shortcomings were identified that highlighted S-CREAM's current weaknesses, potential improvements, and directions for future works.

This is only a first step, which entails that there is a need to apply S-CREAM on more STAs and attack patterns to improve the way it models them and the information it provides. Still, our methodology and its catalogue of AMs are preliminary. Expanding S-CREAM, its lists of antecedents, and the help it provides to



the analyst will guarantee more objectivity in the analysis process and more reliable analysis results.



## Part IV

# Discussion and Conclusion of the thesis work



*The nice thing about having a brain is that one can learn, that ignorance can be supplanted by knowledge, and that small bits of knowledge can gradually pile up into substantial heaps.*

—Douglas Hofstadter, ‘Le Ton beau de Marot’

# 9

## Conclusion

### Contents

---

<b>9.1</b>	<b>Summary</b>	<b>155</b>
<b>9.2</b>	<b>Contributions</b>	<b>156</b>
<b>9.3</b>	<b>Discussion</b>	<b>157</b>
<b>9.4</b>	<b>Directions for future works</b>	<b>160</b>

---

### 9.1 Summary

When one intends to improve the security of a system, the problem of how to protect the system from Socio-Technical Attacks (STAs) is an issue of utmost importance because STAs are cheap, highly effective, and difficult to thwart. But analysing a system for vulnerabilities of a socio-technical nature is a complex matter because these vulnerabilities lie at the junction of the interplay of the system, the user, and the context, and therefore, technical and social aspects should both be considered for an effective analysis.

The research field of usable security has been established as a place where studies on security and usability converge (e.g., [43]). However, this thesis shows that combining other computer sciences and social sciences methods can lead to richer strategies of analysis. One should look at a system and its threats holistically as a place where technical choices can have harmful consequences on the ‘social’ side, and consequently, on the effective system’s security and vice-versa.

Furthermore, focusing only on some parts of the human-related aspects of a system, such as the user interface, is unsatisfactory. Indeed, there are a host of other contextual aspects that should be considered. For instance, the user training, the user goals, the organisation’s security policies, the organisation’s hierarchical

setup, the user's time-constraints, and user physical location can all impact the interactions that the user has with the system.

We stipulate that the best strategy for the analysis of socio-technical systems' security is to study the factors that can influence a user to engage in behaviours that can harm security. Therefore, this thesis tackles socio-technical security analysis by studying how to identify harmful user's behaviours and their influencing factors. In Chapter 2, we stated the following research questions:

### Research Questions

*RQ<sub>1</sub> How can we detect a socio-technical vulnerability in a system?*

*RQ<sub>2</sub> How can we identify in the interactions between a system and its users, the human behaviours that can harm this system's security?*

*RQ<sub>3</sub> How can we identify the factors that foster human behaviours that are harmful to a system's security?*

These research questions pose methodological challenges. Furthermore, they may demand different answers when stated in front of different audiences, i.e., researchers and practitioners, because of the different goals, interests, and capabilities these audiences may have in their respective practices of security analysis. For the resolution of these research questions we identified the following objectives for this thesis:

### Objectives

*O<sub>1</sub> What form should a framework that intends to identify harmful behaviours for security, and to investigate the factors that foster their occurrence take?*

*O<sub>2</sub> What form should a semi-automatic, or tool-assisted methodology for the security analysis of socio-technical systems take?*

We adopted the two following approaches to tackle these objectives:

- (a) testing hypotheses about potential factors by running experiments
- (b) identifying the factors that have helped STAs to succeed in the past through Root Cause Analyses (RCAs).

## 9.2 Contributions

The thesis reaches the aforementioned objectives through the two following contributions:

**Addressing  $O_1$ : STEAL, a framework for socio-technical security analysis.**

In Part II, we presented a framework that we call the Socio-TEchnical Attack AnaLysis (STEAL) framework. It provides a common ground for an interdisciplinary approach towards security analysis. This common ground is a reference model based on Bella and Coles-Kemp’s *concertina model* [16] that allows security researchers to describe a system along with its users and its context in order to identify and investigate potential attacks and defences. STEAL addresses Objective  $O_1$  by allowing security researchers to use computer sciences and social sciences methods sequentially to identify what we call *critical actions*, or actions that can undermine a security property, and to formulate and test hypotheses about the factors that foster an insecure behaviour at these *critical action points*.

**Addressing  $O_2$ : S-CREAM, an RCA for socio-technical security.**

In Part III, we presented a methodology that we call Cognitive Reliability and Error Analysis Method for Socio-Technical security (S-CREAM). It is inspired from an RCA technique used in the safety field called Cognitive Reliability and Error Analysis Method (CREAM) [91]. The S-CREAM RCA methodology offers the following features, which when used in combination, answer Objective  $O_2$ : (1) it implements a *Retrospective Analysis* to pinpoint the factors, amongst those available from a catalogue of human-related factors, that contributed to the success of an attack, (2) it allows for the constitution of a knowledge database of factors that are known to have facilitated attacks in the past, and (3) it provides a tool-supported operational procedure —S-CREAM’s *Security Analysis*— to semi-automatically detect if these factors are applicable on a specific system when the threat model is given.

## 9.3 Discussion

We now discuss how the thesis’s contributions allow to answer the research questions, and how they sustain accurate and objective socio-technical security analyses for each community of users.

As demonstrated in Chapter 4 and Chapter 7, both the STEAL framework and the S-CREAM methodology can be used to identify socio-technical vulnerabilities. This addresses the first research question  $RQ_1$ .

STEAL and S-CREAM do not share the same approach, and they do not have the same purpose. Indeed, STEAL is geared towards the close inspection of a system by a security researcher. On the other hand, S-CREAM intends to reuse knowledge gained from past research works and past analyses, in the most objective and automated way as possible, for use by security practitioners.

The STEAL framework is more apt for an exploratory approach, supporting the investigations of a security researcher. As shown in Chapter 3, in this framework, the researcher is invited to pinpoint the *critical actions* that could lead to an attack, by addressing  $RQ_2$ . Then the researcher can formulate hypotheses regarding their success and the causes behind it, without other support than his own knowledge and experience. After performing the required experiments to test his hypotheses, the researcher gains insights about the potential STAs. This allows the researcher

to identify subtle socio-technical vulnerabilities with complex causing factors, by addressing  $RQ_3$ .

STEAL offers a framework to support scientific methods, and using STEAL, security researchers are involved in every aspect of the research process. The researchers choose the hypotheses and research methods, and they stay in control of the collection of data, their exploitation, and their interpretation. As a consequence, the security researchers assume responsibility for their assumptions, know the limitations of their analyses, and can prove that their results are up to the scientific standards.

That makes STEAL particularly suitable for research, however, without the need for producing scientific proofs to back up their findings, STEAL is of less appeal to security practitioners. The analysis of a system's security produced by using STEAL can be very accurate if the security researcher has the resources to focus on several *critical action points* and to test several hypotheses. But realistically, researchers can't investigate every possible hypotheses, and they also have to choose between which hypotheses to focus on and which ones to leave for 'future work'. We identified the three following main reasons for hypotheses to be discarded in spite of being worthwhile to the investigation: (1) the researcher can fail to formulate the hypothesis because it is outside his area of expertise, (2) the experiment needed to investigate the hypothesis may be too expensive or impossible for the researcher to realise, and (3) the researcher may give the priority to other hypotheses that appear more likely to produce positive results.

The experiments we reported on in Chapter 4 are good illustrations of these issues. Indeed, we identified several potential hypotheses on Wi-Fi hotspots that are worth investigating, however, we only investigated hypotheses on the *critical actions* we found in the selection phase. The main reasons why the experiments have focus on the low-hanging fruits are extraneous to research itself and are linked to the Ph.D. process. Indeed, if we focused on the simple issues, it was for the pedagogical virtue of the exercise, for the guarantee of being able to produce reliable and exploitable results through experiments of manageable complexity, and for the ability to handle every aspect of experiments as well as to assume control of the process.

Despite having presented STEAL and S-CREAM as having been designed to satisfy the need of one community in particular (researchers for the former, and practitioners for the later), both STEAL and S-CREAM can be used by and be beneficial to both communities.

While we presented S-CREAM as a methodology born from the needs of security practitioners, in reality, its creation is rooted in some of the frustrations we had as security researchers confronting the complexity of using STEAL. Indeed, it is the discrepancy between the complexity of the hypotheses we investigated using STEAL and the amount of time we needed to perform the investigations that originally launched the reflections around the creation of S-CREAM. Studying the effective security of a whole system using STEAL seemed impractical and it highlighted the need for a methodology that, instead of acting as a spotlight on pre-defined hypotheses, could give an overview of the STAs that could affect a system without the burden of the full-featured scientific methods.



Where STEAL and S-CREAM both provide valid answers to the third research question  $RQ_3$  (identify factors), the former provides scientific proofs of the results it produces, and the latter produces a comprehensive overview of all the factors that can potentially be involved in a socio-technical vulnerability. In contrast to the STEAL framework, the S-CREAM methodology removes the burden of selecting potential factors from the analyst. That is to say, instead of using the analyst's knowledge to produce hypotheses, S-CREAM uses a knowledge base of human-related factors extracted from social sciences and a catalogue of Attack Modes (AMs) built from past analyses to perform socio-technical analyses.

This difference of approach between S-CREAM and STEAL gives the analyst the ability to take shortcuts and avoid some researcher-related biases. Indeed, performing a security analysis with S-CREAM is much quicker and cheaper as there is no experiment to run. Second, the analysis is more comprehensive because it systematically offers potential relevant factors to the analyst (akin to the analyst using a check-list to avoid overlooking any factor). Furthermore, it helps avoid some of the biases that the analyst could suffer from, for instance, the confirmation bias (or even the sunk-cost fallacy given the cost of running experiments). Additionally, S-CREAM allows to extend a threat model to support the formal verification of systems to check that while accounting for a socio-technical vulnerability identified through S-CREAM, the system guarantees some security properties.

S-CREAM offers readily usable results for security practitioners without providing traces to the reasons behind these results. It produces a list of potential STAs along with their contributors, and it is the analyst's burden to ponder on whether or not these contributors should be controlled. Thus, S-CREAM needs to be used with discernment, and one should not jump to its conclusions. This remark is critical if the user is a security researcher. Indeed, we don't think researchers can use S-CREAM results 'as is', but instead, should only use S-CREAM in conjunction with STEAL to boost investigations. For instance, in Chapter 4, we presented experiments on the Wi-Fi selection process and the user understanding of the meaning of the different symbols used in the graphical user interface of Wi-Fi network managers. Where user-studies and surveys had been performed to investigate this problem from scratch (generating frustration), preliminary answers could have been readily obtained through a S-CREAM *Security Analysis* to support the design of these studies. S-CREAM proposes additional inputs, allowing to triangulate findings, and thus contributes to the validation of such findings. Furthermore, S-CREAM's findings can also be fed back into STEAL's formal analysis. Such a combination of methods and data sources is an invaluable asset that contributes to the consolidation of the relatively young field of socio-technical security research.

But S-CREAM has some limitations that one needs to be aware of before using it. First, the S-CREAM methodology needs maintenance. Indeed, both the tables inherited from CREAM and the catalogue of AMs require care for the methodology to reach its full potential. As we discussed in Chapter 8, without proper antecedent-consequent tables dedicated to security, the Contributors yielded by S-CREAM's *Security Analysis* are very generic, and sometimes, difficult to understand. Furthermore, the catalogue of AMs requires the analyst to carefully select the Socio-Technical Capabilities (STCs) for the current analysis. Eventually, the catalogue

itself needs to be fed through the analysis of carefully documented attacks to gain in relevance. Clearly, the catalogue that we bootstrapped from Common Attack Pattern Enumeration and Classification (CAPEC) in this thesis is still rudimentary and needs refinements.

In addition to these maintenance needs, S-CREAM has other limitations. One limitation of the catalogue of AMs is that it fails to provide an order or a rough priority hierarchy for determining which AM exerts the most urgency to be mitigated. The catalogue does not provide chains of AMs that could be broken but only individual AM. This forces the analyst to consider remediations on all the AMs, whereas a strategy geared towards breaking chains of AMs or considering attack trees, could be more efficient.

Another limitation of S-CREAM is that the analyst is not necessarily an expert on all aspects upon which S-CREAM can shed light. This shortcoming is even more salient when we consider the possible additions we can make to its tables. For instance, there is a lot of literature on warnings, and how warnings can have a negative impact on user's decisions when not implemented properly. If this literature were to make its way into S-CREAM's tables with new possible antecedents, the analyst would be expected to be able to decide whether the warnings that are presented to the user fulfil their mission.

While security practitioners are the end-users of S-CREAM, the burden of maintaining the methodology does not have to lie on their shoulders. Indeed, security organisations could take on the task to maintain S-CREAM's tables. STEAL could be used by security researchers to investigate new factors and add to the existing pool of knowledge contained in S-CREAM's tables to increase the amount of time saved in subsequent socio-technical analyses and to guarantee scientific accuracy. We believe that it would be beneficial for both communities to create a better STEAL-S-CREAM link as it could embrace security researchers' work on STAs, and ensure security practitioners of the scientific grounding of the results they obtain through S-CREAM.

## 9.4 Directions for future works

The most challenging research direction for future works on S-CREAM will be to design helpers, such as check-lists, to offer guidance to the analyst who performs *Retrospective Analyses* and *Security Analyses*. The potentiality that an analyst chooses a Contributor without fully understanding its meaning and implication should be removed.

Another great challenge is the specialisation of S-CREAM antecedent-consequent tables inherited from CREAM. Providing up-to-date tables of antecedents that reflect the current state of the research on factors that influence security-related behaviour as well as providing the corresponding up-to-date bibliographic references and helpers is important as it will provide S-CREAM great relevance in its field.

Furthermore, S-CREAM only provides a rudimentary description scheme at the time of writing. Carefully crafted schemes with relevant sets of pre-requisites should be developed to further empower S-CREAM *Generalisation* and *Security Analysis* steps.

The most immediate future work concerns the ability that should be given to the analysts to cooperate on S·CREAM's *Retrospective Analysis* and *Security Analysis* steps in *S·CREAM assistant*. Furthermore, sharing schemes, catalogues of AMs, and sets of attacks could prove useful and may be considered for later implementations.

Given the preceding discussion, it appears that S·CREAM yields results that can lead to the improvement of the overall security of a system. We believe that the shortcomings we identified can be fixed, and that by improving S·CREAM's tables, by maintaining its knowledge of security and human-related factors, and by fostering its use and the sharing of experiences, S·CREAM can be a useful addition to a security practitioner's toolbox in every part of the security life cycle. Furthermore, we believe that STEAL and S·CREAM offer security researchers an environment where they can test and utilise their findings and that these tools can constitute a first step towards a real security researchers-security practitioners partnership regarding STAs.





## Complementary social analysis of the TLS validation use case

We introduce a potential social study that spans from  $UI_A$  to  $S_A$  of the TLS use case. We draft the study of a potential semantic attacks [153][106] on Web Browser's users when validating TLS certificate, where users are asked to choose instead of the Web Browser.

We focus, in particular on a Google Chrome user having to choose whether or not trusting a certificate. We don't contextualise the framework in a research process, but we can instantiate it to describe an attack scenario. We know that users already ignore 60% of Interstitial Warnings (IW) in Google Chrome [119] and this rate may increase if an Attack changes the user's state just before he makes a choice among the options  $UI_A$  offers him. By ignoring, we mean that users prefer to choose the option 'Proceed anyway' and then store the self-signed certificate, over 'Go back to safety' (equivalent to the back action) and closing the tab. An Attack controlling the user interface (i.e., Man-in-The-Browser) can send some specially crafted information (i.e., the payload) in order to force the user (BCP) to deviate from the prescribed (i.e., secure) behaviour. In Figure 3.5 we see that the attack strikes in the  $P_A$ .  $UI_A$  interaction. We consider the application 'Warning' being all the 'Computer' layers until  $UI_A$  and the user being all the 'Human' layers until  $P_A$ .

Here we choose to convey the payload by a fake IW which will be shown before the genuine one. The payload aims at misleading the user in interpreting a self-signed certificate as SELF-signed certificate, that is 'certificate signed by S.E.L.F.', where SELF is a new certification authority yet unknown by the user's browser, but introducing itself as trusted in the text of the IW. This introduces a polysemy on the word 'self' that may lead the user to misinterpret the meaning of the word (called equivocation fallacy e.g. 'The sign said 'fine for parking here', and since it was fine, I parked here.').

As presented by the Sequence Diagram in Figure A.1, the fake IW only offers a ‘continue’ action that leads to the genuine one. The genuine warning offers the option to bring the user back to the fake interstitial warning (loop) or to store the self-signed certificate (then the attack succeeds). The only way for the user to escape this loop and make the attack fail is to hit the browser’s close button at any time or to hit the back button when he is on the fake IW.

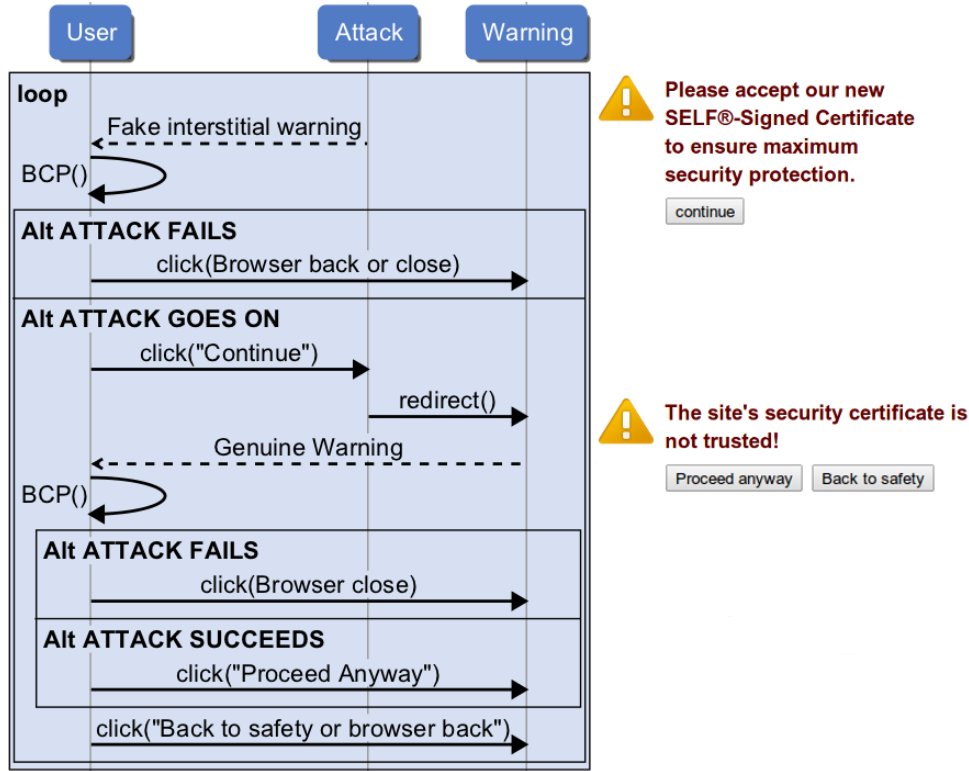


Figure A.1: Sequence diagram for the studied attack scenario. We use alternatives this way to emphasise the needed actions to make the attack fail or succeed. The possible use of the Context  $C_A$  as a defence is not represented here for the sake of simplicity.

The context  $C_A$  does not actively play a role in this attack scenario but could be used by a defence (see the empty arrow in Figure 3.5). For instance, in addition to the explanations the Genuine Warning given to the user, a link to some education / information material could be proposed. This could affect the user’s state and change his following behaviour, hopefully to one that tends to be more secure.

To test whether or not this semantic attack could indeed work, we imagine an experiment where a set of participants are placed in this situation and the attack launched. A fake IW would be designed from the Chromium’s source code and the attack implemented in a tool like BeEF [5] (and has actually been implemented in BeEF in a module called ‘Chrome cert beggar’). To test any hypothesis about how people resist such misleading inputs we need to instantiate the whole research process (e.g., to decide how to launch the attack, what and how to observe, what to ask users afterwards, etc.) and to run the corresponding experiments. Then we could observe the patterns that lead users to resist or to fall for the attack and try

---

to explain those patterns with the analysis of the qualitative investigations. Such an experiment would need authorisation from an ethical committee, compliance with a legal framework, and with ethical requirements (APA), before being set.

This draft of a social study of the TLS is presented here for the sake of completeness. It worth noting that this toy study further shows that STEAL can support both social sciences and computer sciences methods on a common model.





# B

## Questionnaire used for testing the influence of graphical cues in the selection of Wi-Fi networks

In the following, we present an example of participation to the study about the influence of graphical cues in the selection of Wi-Fi networks described in Chapter 4 (see Section 4.3.2). In this particular case, the participant is assigned to the condition in which he is given a password. As explained in Chapter 4, rounds of network selection are randomised, and one round is randomly reintroduced to check the consistency of the participant's answers; in this particular case, rounds are displayed in the following order: 2nd, 3rd, 1st, 3rd (reintroduced), and 4th.

## Thank you for accepting this HIT!

By clicking the following URL link, you will be taken to the survey, including complete instructions and an consent agreement.

**Warning:** Please disable pop-up blockers before continuing.

Begin survey

Figure B.1: This is the questionnaire's landing page that the participant reaches when picking our HIT from the amazon mechanical turk HIT list.

## Consent form.

### Purpose of the Study and Related Information

You have been invited to take part in a research study on the way people interact with wireless networks; in this study we will collect data about:

- Some basic demographic information about you,
- your stated choices regarding wireless networks,
- your answers to a questionnaire, and
- timestamps of all your interactions with the system.

We adhere to international standards regarding data collection, privacy and ethics e.g., those provided by the American Psychological Association and CNPD (Commission Nationale pour la Protection des Données - Luxembourg). You have the right to end your participation in the study at any point and to ask for your data to be removed, but doing so will prevent you from claiming your mTurk HIT.

### Use of data

We intend to use the data for academic research purposes only **AND** peer-reviewed publications; we will however **NOT** identify you individually in any reports, publications or to external parties without your explicit agreement. For example your name will never be published in connection with any data collected, instead only anonymous and aggregated data will be used. For the purposes of this research project, the data will be stored according to CNPD recommendations.


### HIT reward eligibility

You will be awarded \$0.90 USD for completing this HIT. However, the survey contains answer-consistency checks and timing-checks to verify that the answers were not provided by an automation tool.

**You are allowed to take this survey only once.**

### Contact information

Contact email : [jean-louis.huynen@uni.lu](mailto:jean-louis.huynen@uni.lu)

 Print a copy of this

Do you understand and agree to these terms?



I agree

No thanks, I do not want to do this HIT

Figure B.2: First, we ask for the participant's consent to participate to the study.

Thank you for accepting to take this HIT. 1/11

Before beginning, we need to perform a technical check in order to ensure your system displays exactly what we need you to see.

	Yes	No
Do you see a picture of a thumbs up between those brackets: [  ]?	<input checked="" type="radio"/>	<input type="radio"/>
Do you see a picture of a thumbs up between those brackets: [  ]?	<input checked="" type="radio"/>	<input type="radio"/>

[Continue →](#)

Figure B.3: Then, we check whether the participant is technically able to participate.

Please answer the following questions. 2/11

What is your age?

Are you of Hispanic, Latino, or Spanish origin?  Yes  No

Please choose one or more races that you consider yourself to be:

- White
- Black or African American
- American Indian or Alaska Native
- Asian
- Native Hawaiian or other Pacific Islander
- Other

What is your gender?

What is the highest degree of education that you have completed?

What is your occupation?

How comfortable are you with ICT (Information and Communication Technologies):

[Continue →](#)

Figure B.4: In this first part of the collection of data, we retrieve socio-demographic information from the participants.

Instructions. 3/11

In the next pages, you will be presented with a description of a situation where you need to use a Wi-Fi network (wireless network). You are asked to click on the Wi-Fi network you would choose.

The password is: SURVEY

[Continue →](#)

Figure B.5: This page gives instructions to the participant regarding the survey. Furthermore, the participant is randomly assigned to a condition (with password / without a password), in this particular case, the participant is given a password.

Please select a Wi-Fi network. 4/11

---

I am sitting in a coffee shop with some friends. As they want to go for dinner later, I use my smartphone to check for a good restaurant. Unfortunately, there is no 3G/4G network available, so I have to use an available Wi-Fi network instead.

Which network would you choose?

	h04sj	📶
	mbt1m	📶
	0dwex	📶
	e2oje	📶

Figure B.6: This is the 1st choice out of 5. In this particular case, the 1st choice is a 2nd round. The participant is assigned to a scenario displayed on the top of the page. Once defined, the same scenario is used for all the subsequent choices this participant is asked to make.

Please select a Wi-Fi network. 5/11

---

I am sitting in a coffee shop with some friends. As they want to go for dinner later, I use my smartphone to check for a good restaurant. Unfortunately, there is no 3G/4G network available, so I have to use an available Wi-Fi network instead.

Which network would you choose?

	zo4qr	📶🔒
	8w6dm	📶🔒
	1bpbi	📶
	p95g6	📶

Figure B.7: This is the 2nd choice out of 5. In this particular case, the 2nd choice is a 3rd round.

Please select a Wi-Fi network. 6/11

I am sitting in a coffee shop with some friends. As they want to go for dinner later, I use my smartphone to check for a good restaurant. Unfortunately, there is no 3G/4G network available, so I have to use an available Wi-Fi network instead.

Which network would you choose?





ab6vk	
f8y9g	
6ka6d	
8hhyh	

Figure B.8: This is the 3rd choice out of 5. In this particular case, the 3rd choice is a 1st round.

Please select a Wi-Fi network. 7/11

I am sitting in a coffee shop with some friends. As they want to go for dinner later, I use my smartphone to check for a good restaurant. Unfortunately, there is no 3G/4G network available, so I have to use an available Wi-Fi network instead.

Which network would you choose?





gqzo3	
x59ki	
fki62	
rdpps	

Figure B.9: This is the 4th choice out of 5. In this particular case, the 4th choice is a 3rd round. This 3rd round has been reintroduced randomly to check the consistency of the participant's answers.

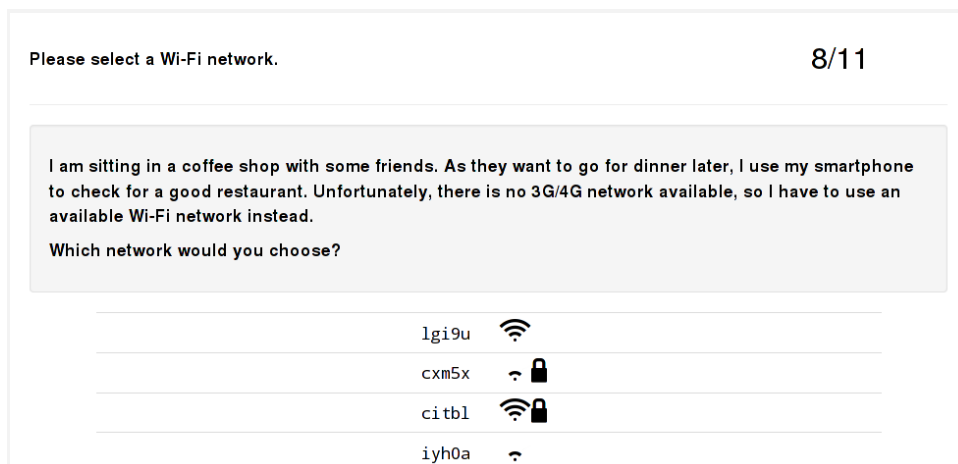



Figure B.10: This is the last choice. In this particular case, the 5th choice is a 4th round. In a 4th round, the networks names appear one after the other, each subsequent addition is delayed by 200ms.


Appendix B. Questionnaire used for testing the influence of graphical cues in the selection of Wi-Fi networks

9/11


Now, please tell us about your understanding of the following symbols. Indicate for each symbol how much you agree with the listed properties.

I would say that a  means:

	Not at all	Partially	Mostly	Completely
Private	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confidential	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Speed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good Signal Strength	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Protected	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Bandwidth	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encrypted	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I would say that a  means:

	Not at all	Partially	Mostly	Completely
Private	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confidential	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Speed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good Signal Strength	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Protected	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Bandwidth	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encrypted	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I would say that a  means:

	Not at all	Partially	Mostly	Completely
Private	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confidential	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Speed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good Signal Strength	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Protected	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Bandwidth	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encrypted	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

[Continue →](#)



Figure B.10: (previous page) Here, we ask the participants to rate on 4-points likert scales, their understanding of the extent to which they agree that each of the 3 visual cues corroborate in meaning with 4 words related to ‘Encryption’, and ‘QoS’.

Please answer each of the following questions. 10/11

---

**When I choose a wireless network:**

	Yes	No
I don't want to enter a password.	<input type="radio"/>	<input type="radio"/>
I choose whatever works.	<input type="radio"/>	<input type="radio"/>
I avoid the Wi-Fi networks with a  .	<input type="radio"/>	<input type="radio"/>

**I think that:**

	Yes	No
Computer security threats are overrated.	<input type="radio"/>	<input type="radio"/>
I answer randomly and I should not be paid.	<input type="radio"/>	<input type="radio"/>
I'm lucky; I never have security problems.	<input type="radio"/>	<input type="radio"/>
Only gullible people fall for Internet scams.	<input type="radio"/>	<input type="radio"/>
The service provider takes care of security.	<input type="radio"/>	<input type="radio"/>
I take additional security measures to protect myself.	<input type="radio"/>	<input type="radio"/>
There is always enough signal strength for what I do online.	<input type="radio"/>	<input type="radio"/>
Between a  and a  ,there is no big difference.	<input type="radio"/>	<input type="radio"/>

[Continue →](#)

Figure B.11: After having investigated the participant’s understanding of the graphical cues, we ask complementary questions about the participant’s attitude and belief regarding the participant use of Wi-Fi networks.

Please answer each of the following questions. 11/11

---

### Networks and Devices

I use Wi-Fi Networks: Sometimes ▾

I use Ethernet Networks: Often ▾

I use VPN or other tunneling methods: I always use a VPN ▾

Please select the devices that you use at least once a day (Hold the CTRL key to select several options):

▾

- (dumb) Cell Phone
- Smartphone
- Personal Computer
- Tablet

	Yes	No
Have you ever set up a Wi-Fi Network?	<input checked="" type="radio"/>	<input type="radio"/>
Is the Wi-Fi network more secure when a  is displayed next to the network name ?	<input checked="" type="radio"/>	<input type="radio"/>
Is it secure to do online banking on a Wi-Fi network when a  is displayed next to the Wi-Fi network name?	<input checked="" type="radio"/>	<input type="radio"/>
	True	False
Wi-Fi network names are verified by a Certification Authority.	<input type="radio"/>	<input checked="" type="radio"/>
When a  is presented next to the Wi-Fi network name, everything you do online through this connection is secure.	<input type="radio"/>	<input type="radio"/>
When you are selecting from a list of available Wi-Fi networks, the one listed on top is the most secure.	<input type="radio"/>	<input type="radio"/>
Wi-Fi network names are protected and can not be spoofed (impersonated).	<input type="radio"/>	<input type="radio"/>
When a  is displayed close to a Wi-Fi network name, it means "Do not enter".	<input type="radio"/>	<input type="radio"/>
It is not secure to use the last Wi-Fi network in a list of available Wi-Fi networks.	<input type="radio"/>	<input type="radio"/>

Finish →

Figure B.12: In this last page of collection of data, we continue our investigation of the participant's beliefs, and we ask the participant about his connectivity habits.

---

## Thank you!

Thank you for your participation in our study! Your anonymous data makes an important contribution to our understanding of human behaviour regarding computer security.

If you have any questions about this research, or are interested in the results, you may contact [jean-louis.huynen@uni.lu](mailto:jean-louis.huynen@uni.lu).

Complete HIT! →

Figure B.13: Finally, we thank the participant and redirect him to the amazon mechanical turk website.



# C

## Pilot study performed to prepare the questionnaire presented in Appendix B

In the following, we give additional details on the pilot-study that we conducted in order to design the questionnaire that investigates the influence of graphical cues in the selection of Wi-Fi networks (see Chapter 4 and Appendix B).

### C.1 The vignettes

As explained in Section 4.3.2, we created and tested 3 vignettes, or candidate scenarios, for each scenario. Table C.1 lists the 3 vignettes that intend to convey the lack of need for ‘Encryption’ and ‘QoS’ (Quality of Service); Tables C.2 lists the vignettes that intend to convey the lack of need for ‘Encryption’, and the need for ‘QoS’; Table C.3 lists the vignettes that intend to convey the need for ‘Encryption’, and the lack of need for ‘QoS’; and finally, Table C.4 lists the vignettes that intend to convey the need for ‘Encryption’, and the need for ‘QoS’.

### C.2 The questionnaire

The questionnaire used in the pilot-study has the same structure (and uses the same tools and settings) as the questionnaire presented in Appendix B. First, we greet the participant and ask for his consent to participate, then we give the participant some instructions (see Figure C.1) and collect diverse socio-demographic information. Then, comes the part that is specific to the pilot-study: the rating of the different vignettes on 4 points likert scales (‘Not necessary’, ‘Not important’, ‘Important’, and ‘Mandatory’). As we show in Figure C.2, the participant rates on different likert scales how much the Wi-Fi network mentioned in the displayed candidate scenario requires each listed property. 6 properties are related to ‘En-

Table C.1: Vignettes that intend to convey the lack of need for ‘Encryption’ and ‘QoS’.

Vignette	Intended meaning		Displayed text
	Encryp.	QoS	
$A_1$	0	0	I am sitting in a book shop. As I am alone and have some time to spare, I want to use my smartphone to check my social network accounts for news about my friends. Unfortunately, there is no 3G/4G network, so I use an available Wi-Fi network instead.
$A_2$	0	0	I am sitting in a coffee shop with some friends. As they want to go for dinner later, I use my smartphone to check for a good restaurant. Unfortunately, there is no 3G/4G network available, so I have to use an available Wi-Fi network instead.
$A_3$	0	0	I am in a bar and have had a couple of drinks too many. Thus, I need to find the nearest hotel in walking distance with a vacant room. Unfortunately, there is no 3G/4G network, so I use an available Wi-Fi network instead.

Table C.2: Vignettes that intend to convey the lack of need for ‘Encryption’, and the need for ‘QoS’.

Vignette	Intended meaning		Displayed text
	Encryp.	QoS	
$B_1$	0	1	I just got off the bus, and I’m heading to a job interview. I check my smartphone to figure out how to get to my destination, but there is no 3G/4G and the smartphone is unable to display the map of the current location. I thus decide to try an available Wi-Fi network to get some connectivity.
$B_2$	0	1	I am a graphic designer intending to show my latest work to some of my friends. Since the 3G/4G connection is failing to retrieve the files, which are rather big, I decide to try an available Wi-Fi network to get some connectivity.
$B_3$	0	1	I am a movie producer on the road. My assistant calls me because he needs feedback on an HD movie trailer that a client needs tonight. I only have my smartphone and even though it is a relatively big file I need to download it. Unfortunately, there is no 3G/4G. I thus decide to try an available Wi-Fi network to get some connectivity.

Table C.3: Vignettes that intend to convey the need for ‘Encryption’, and the lack of need for ‘QoS’.

Vignette	Intended meaning		Displayed text
	Encryp.	QoS	
$C_1$	1	0	I am waiting at a bus stop and I need to verify whether the check I deposited yesterday has been cleared. I need to use the bank’s application on my smartphone to check the bank account’s balance, but unfortunately there is no 3G/4G. I thus decide to try an available Wi-Fi network to get some connectivity.
$C_2$	1	0	I am in a movie theater waiting for the movie to begin. As I have a bit of extra time, I decide to order my groceries online for the week. Unfortunately, there is no 3G/4G. I thus decide to try an available Wi-Fi network.
$C_3$	1	0	I am a Medical Doctor attending a conference and get contacted by a colleague in order to provide advice on a patient. In order to understand the case, I have to download the patient’s file from our private practice server. Unfortunately, there is no 3G/4G so I decide to try an available Wi-Fi network to get some connectivity.

ryption’ (Private, Secret, Masked, Protected, Encrypted, and Confidential), and 6 properties are related to ‘QoS’ (Fast, Responsive, First-Class, Good signal strength, High-bandwidth, and High-speed). The participant repeats this rating operation for the 12 vignettes. Then, we ask the participant several questions about his connectivity habits, and his beliefs about Wi-Fi networks (see Figure C.3). Finally, we debrief and thank the participant before redirecting him to the amazon mechanical turk web site.

## Instructions

You will be shown twelve scenarios. In each scenario you will be asked to imagine yourself in a specific situation that calls for the use of a Wi-Fi network to perform a specific task, and you will be asked to rate a set of Wi-Fi properties for that particular scenario using the following scale: "Not Necessary", "Not Important", "Important", "Mandatory".

When you are ready to begin, click on the "Begin survey" button.

Begin survey →

Figure C.1: This page gives instructions to the participant regarding the survey.

Table C.4: Vignettes that intend to convey the need for ‘Encryption’ and ‘QoS’.

Vignette	Intended meaning		Displayed text
	Encryp.	QoS	
$D_1$	1	1	I am rushing to meet a client to sign a contract. While hurrying there, I receive an SMS from a co-worker: ‘The boss changed the contract and put it on the company’s Intranet and wants you to use the updated version instead’. I need to get the file rapidly, but unfortunately I can’t access the company’s Intranet from my tablet. I ask my colleague to send me the private document via email, which I can read from the tablet. There is no 3G/4G, and thus I decide to use an available Wi-Fi network to read my mail, retrieve and save the attached file. This particular file is huge.
$D_2$	1	1	I am a government official staying at an hotel. I scheduled an international online meeting. I planned to use the hotel’s Wi-Fi network but the hotel’s Wi-Fi proved unreliable when I called my family earlier to test the connection. There is no 3G/4G network, so I decide to go somewhere else to find an available Wi-Fi network.
$D_3$	1	1	I am a highly regarded technology blogger and was invited among a selected few to attend a virtual pre-release video presentation on the next hottest gadget. This is a unique opportunity to be among the first to report on this new item, which has the potential to attract millions of hits to my website. As I am stuck in traffic and there is no 3G/4G network, I decide to pull over to try an available Wi-Fi network.

### C.3 Choosing the best vignette to convey the scenario’s meaning

To choose the vignette that will convey the best the intended meaning, we build boxplots corresponding to the participants’ answers for 2 categories of ratings: Needed, and Not needed. Not needed and Needed are the counts of participant ratings respectively of the ‘Not necessary’ and ‘Not important’, and ‘Important’ and ‘Mandatory’. As an example, Figure C.4 shows the boxplots corresponding to the vignettes that intend to convey that ‘Encryption’ and ‘QoS’ are not needed. We search for a vignette for which participants preferred the Not needed category ratings over the Needed category ratings—for both dimensions. In this particular example,  $A_1$  is not the best candidate because, for the ‘Encryption’ dimension,  $A_2$  and  $A_3$  are evidently better (their counts of Not needed are higher, and their counts of Needed are lower). As the difference between  $A_2$  and  $A_3$  on this dimension is less evident, we perform several Mann-Whitney-Wilcoxon tests (the data do not satisfy the t-test’s



I am a graphic designer intending to show my latest work to some of my friends. Since the 3G/4G connection is failing to retrieve the files, which are rather big, I decide to try an available Wi-Fi network to get some connectivity.

For this scenario, I would require the Wi-Fi network to have the following properties:

Property	Not Necessary	Not Important	Important	Mandatory
Private	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confidential	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Speed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Secret	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Responsive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Masked	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good Signal Strength	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Protected	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
First-Class	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encrypted	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High-Bandwidth	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Scenario 1/12 Next scenario →

Figure C.2: This page asks the participant to rate, for the described scenario, how much each property is required by the Wi-Fi network. The participant is asked to perform this task for the 12 vignettes.

pre-conditions). It appears that  $A_2$ 's Needed category count is significantly lower than  $A_3$ 's ( $p=0.01$ ), and that  $A_2$ 's Not needed category count is significantly higher than  $A_3$ 's ( $p=0.01$ ). We do not have a significant difference on the 'QoS' dimension, but when grouping by gender, we identify a gender effect that we want to avoid in  $A_3$ . Indeed, we find out that the women who participated to the study significantly tend to rate the  $A_3$  vignette more as requiring the 'QoS' property compared to men, and this can be problematic for the subsequent questionnaire. In  $A_2$ , the only gender effect we find is that the men who participated to the study tend to rate the 'QoS' as being Not needed significantly more than women, which is a difference we need to be aware of when doing the analysis of the subsequent questionnaire. Therefore we choose the vignette  $A_2$  to convey the lack of need of 'Encryption' and 'QoS'.

## Follow-up questionnaire

Congratulations, you finished the survey! Please now respond to the following questions to complete your HIT.

### Networks and Devices

I use Wi-Fi Networks:

I use Ethernet Networks:

Please select from the list of devices those you use at least once a day (Hold the CTRL key to select several options):

- (dumb) Cell phone
- Smartphone
- Personal Computer
- Tablet

### Yes or No

Have you ever set up a Wi-Fi Network?

Have you ever fallen victim to an Internet Scam?

Is the network is more secure when the signals strength is better?

Is it secure to do online banking on a Wi-Fi network when the signal strength is bad?

### True or False

Wi-Fi network names are verified by a certification Authority.

When a Padlock is present next to the Wi-Fi network name, everything you do online through this connection is secure.

When you are selecting from a list of available Wi-Fi networks, the one listed on top is the most secure.

Wi-Fi network names are protected and can not be spoofed (impersonated).

The Padlock sign **only** means "Do not enter".

It is not secure to use the last Wi-Fi network in a list of available Wi-Fi networks.

[Continue →](#)

Figure C.3: This page gathers information regarding the participant's connectivity habits, and the participant's beliefs about Wi-Fi networks.

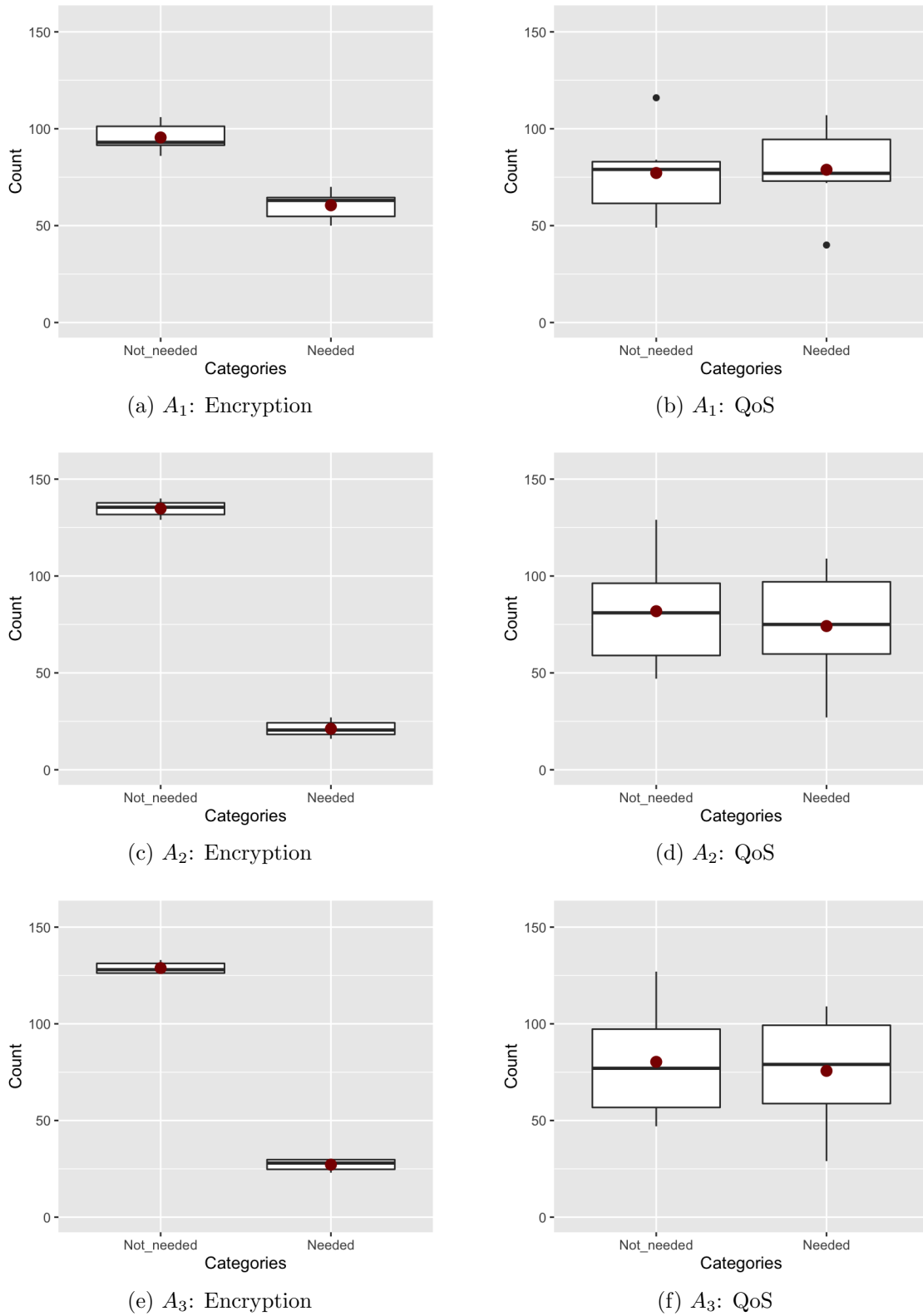


Figure C.4: Boxplots depicting the participants' ratings corresponding to the vignettes that intend to convey the lack of need for 'Encryption' and 'QoS'.



# D

## Questions used in the study on the influence of the context and trust on Wi-Fi network selection

In this appendix, we present the questions asked to the participants who were part of the study that investigated the influence of the network names (see Section 3.3) and trust (see Section 4.3.1.1) on user Wi-Fi selection. The questionnaire has a set of common questions, first about socio-demographic information, second about the participant's baseline preferences regarding the network names used in the study. The questionnaire has 2 other rounds of questions that correspond to 2 conditions to which the participants were randomly assigned: the context condition, and the trust condition. The questionnaire was available in English, German, and French.

### D.1 Questions common to both conditions

The survey begins with the gathering of socio demographic information. We collect: age, gender, education (high school, bachelor's degree, master's degree, phd, other), and occupation. Furthermore, we ask if the participant is living or working in Luxembourg (the location where the Wi-Fi network names were collected). We also collect information regarding the participant computer literacy. The participant answers on a 5 points likert scale (not at all comfortable, not very comfortable, neutral, comfortable, very comfortable) to the question 'I feel comfortable using Information Technology (computers, smartphones, tablets...)', and on a 5 points likert scale (very good, good, average, not very good, not at all good) to the question 'I would describe my IT skills to be'.

Then every participant is presented the following instructions:

Please rate each wireless network listed below, in terms of **preference**, and avoid selecting the option ‘neutral’ unless you cannot rate the name at all.

These instructions are presented along with the list of network names (eduroam, uni-visitor, uni-student, wifi\_unilu, hotcity, Hotel.le.Place.D’Armes, Cafe.de.Paris, secured\_hotspot, secure\_wifi\_BelleEtoile, free\_wifi\_BelleEtoile, Maroquinerie\_Kirchberg, and free\_AP), and the corresponding 5 point likert scales (not at all preferred, not very preferred, neutral, preferred, most preferred). Figure D.1 shows an illustration of the actual presentation of this question.

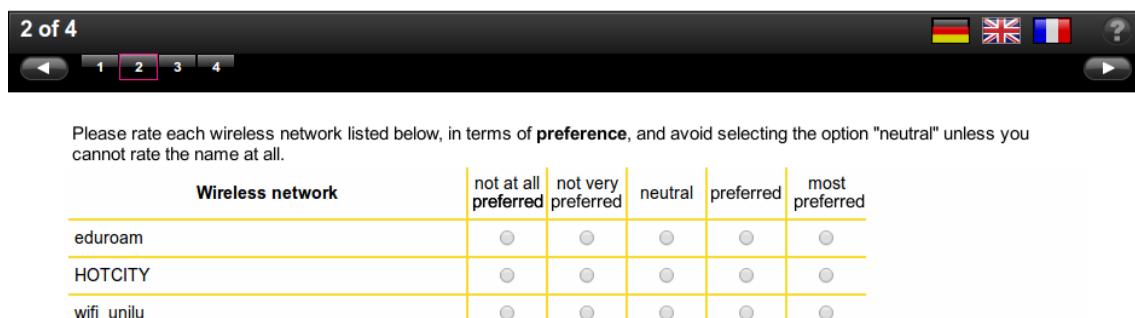


Figure D.1: Illustration of the actual presentation of the online questionnaire, when asking for the participant’s preferences regarding network names, in English. This question is presented several times to the participants throughout the questionnaire, with different instructions and likert scale ratings.

Then the participant is asked to provide an explanation for why he rated some names higher than others in an open text field.

## D.2 Trust condition

In the trust condition, the participant is only presented one additional page with two questions. The participant is asked to rate the network names regarding trust on 5 points likert scales:

Please rate each wireless network listed below, in terms of **trust** (i.e. if you perceive some risk or threat associated with the name or if you consider it to be trusted) (not at all trusted, not very trusted, neutral, trusted, highly trusted), and avoid selecting the option ‘neutral’ unless you cannot rate the name at all.

Then the participant is asked to provide an explanation for why he rated some names higher than others in an open text field.

## D.3 Context condition

In the context condition, the participant is presented 4 successive questions in which the participant is asked to rate the network regarding his preferences. Only, each time the location is different.

Imagine that you are located at the **University of Luxembourg**. Please rate each wireless network listed below, in terms of **preference** (not at all preferred, not very preferred, neutral, preferred, most preferred), and avoid selecting the option ‘neutral’ unless you cannot rate the name at all.

Imagine that you are located at the **City Center in Luxembourg (e.g. Place D’Armes)**. Please rate each wireless network listed below, in terms of **preference** (not at all preferred, not very preferred, neutral, preferred, most preferred), and avoid selecting the option ‘neutral’ unless you cannot rate the name at all.

Imagine that you are located at the **shopping mall (e.g. Belle Etoile)**. Please rate each wireless network listed below, in terms of **preference** (not at all preferred, not very preferred, neutral, preferred, most preferred), and avoid selecting the option ‘neutral’ unless you cannot rate the name at all.

Imagine that you are located at the **Hospital (e.g. Kirchberg Hospital)**. Please rate each wireless network listed below, in terms of **preference** (not at all preferred, not very preferred, neutral, preferred, most preferred), and avoid selecting the option ‘neutral’ unless you cannot rate the name at all.

After each question, the participant is asked to provide an explanation for why he rated some names higher than others in an open text field.





# Bibliography

- [1] 3GPP Technical Specification 24.312 Access Network Discovery and Selection Function (ANDSF) Management Object (MO), December 2013.
- [2] I. Abdelhalim, S. Schneider, and H. Treharne. An integrated framework for checking the behaviour of fUML models using CSP. *International Journal on Software Tools for Technology Transfer*, 2012.
- [3] A. Adams and M. A. Sasse. Users Are Not the Enemy. *Comm. ACM*, 42:40–46, 1999.
- [4] D. Akhawe and A. P. Felt. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *in Proc. of the 22nd USENIX Security Symposium, August 14-16, 2013, Washington, DC, USA*, 2013.
- [5] W. Alcorn. BeEF: The Browser Exploitation Framework. Available at <http://beefproject.com/>.
- [6] S. Alfawaz, K. Nelson, and K. Mohannak. Information security culture: A behaviour compliance conceptual framework. In *Proceedings of the Eighth Australasian Conference on Information Security - Volume 105, AISC '10*, pages 47–55, Darlinghurst, Australia, Australia, 2010. Australian Computer Society, Inc.
- [7] A. Algarni, Y. Xu, T. Chan, and Y. Tian. Social engineering in social networking sites: Affect-based model. In *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*, pages 508–515. IEEE, 2013.
- [8] Wi-Fi Alliance. Wi-Fi CERTIFIED Passpoint: A new program from the Wi-Fi Alliance to enable seamless Wi-Fi access in hotspots., June 2012. <http://www.wi-fi.org>.
- [9] R. Anderson and T. Moore. Information Security Economics - and Beyond. In *DEON '08: Proceedings of the 9th international conference on Deontic Logic in Computer Science*, volume 5076, pages 1–26. Springer, July, 15-18 2008.
- [10] R. J. Anderson. *Usability and Psychology*, chapter 2. Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, 2008.
- [11] APM Group Ltd. ITIL Home, 2012. <http://www.itil-officialsite.com/>.
- [12] I. Arce. The weakest Link Revisited. *Security Privacy, IEEE*, 1(2):72 – 76, mar-apr 2003.

- [13] S. Band, D. Cappelli, L. Fischer, A. Moore, E. Shaw, and Trzeciak R. Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis. Technical Report CMU/SEI-2006-TR-026, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2006.
- [14] J. E. Bardram. The trouble with login: on usability and computer security in ubiquitous computing. *Personal and Ubiquitous Computing*, 9(6):357–367, 2005.
- [15] A. Beautement, M. A. Sasse, and M. Wonham. The compliance budget: managing security behaviour in organisations. In *Proc. of NSPW 08, Lake Tahoe, California, USA, September 22-25, 2008*, pages 46–58. ACM, 2008.
- [16] G. Bella and L. Coles-Kemp. Layered analysis of security ceremonies. In *Information Security and Privacy Research: 27th IFIP TC 11 Information Security and Privacy Conference, SEC 2012, Heraklion, Crete, Greece, June 4-6, 2012. Proceedings*, pages 273–286. Springer, 2012.
- [17] G. Bella, P. Curzon, and G. Lenzini. Service security and privacy as a socio-technical problem. *Journal of Computer Security*, 25(5):563–585, 2015.
- [18] G. Bella, R. Giustolisi, and G. Lenzini. A Socio-Technical Understanding of TLS Certificate Validation. In *Proc. of 7th IFIPTM 2013*. IFIP, 2013.
- [19] G. Bella, R. Giustolisi, and G. Lenzini. Socio-Technical Formal Analysis of TLS Certificate Validation in Modern Browsers. In *Proc. of PST 2013*. IFIP, 2013.
- [20] M. Blaze. *Toward a Broader View of Security Protocols*, volume 3957, pages 106–120. Springer.
- [21] R. L. Boring. Fifty Years of THERP and Human Reliability Analysis. *Proceedings of PSAM11*, 2012.
- [22] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.
- [23] B. M. Bowen, R. Devarajan, and S. Stolfo. Measuring the human factor of cyber security. In *Technologies for Homeland Security (HST), 2011 IEEE International Conference on*, pages 230–235, Nov 2011.
- [24] J. Brodtkin. NFL to block mobile streaming video in super bowl stadium, January 2014. <http://arstechnica.com/information-technology/2014/01/nfl-to-block-mobile-streaming-video-in-super-bowl-stadium>.
- [25] J. S. Brown, A. Collins, and P. Duguid. Situated cognition and the culture of learning. *Educational Researcher*, 18(1):32–42, 1989.
- [26] J. Brumfield. 2015 Data Breach Investigations Report. Technical report, Verizon, 2015.

- 
- [27] P. C. Cacciabue. *Guide to Applying Human Factors Methods - Human Error and Accident Management in Safety-Critical Systems*. Springer, 2004.
- [28] L. J. Camp. Mental models of privacy and security. *IEEE Technology and Society Magazine*, 28(3):37–46, 2009. cited By 23.
- [29] M. C. Carlos, J. E. Martina, S. Catarina, G. Price, and R. F. Custódio. An Updated Threat Model for Security Ceremonies. In *Symposium on Applied Computing*, pages 1836–1843, 2013.
- [30] M. C. Carlos and G. Price. Understanding the weaknesses of human-protocol interaction. In Jim Blyth, Sven Dietrich, and L. Jean Camp, editors, *Financial Cryptography and Data Security*, volume 7398 of *Lecture Notes in Computer Science*, pages 13–26. Springer Berlin Heidelberg, 2012.
- [31] C. Castelfranchi and R. Falcone. *Trust Theory: A Socio-Cognitive and Computational Model*. Wiley, 2010.
- [32] A. Chapanis. *Research techniques in human engineering*. Johns Hopkins Press, 1959.
- [33] T. Chenoweth, R. Minch, and S. Tabor. Wireless insecurity: examining user security behavior on public networks. *Commun. ACM*, 53(2):134–138, February 2010.
- [34] R. B. Cialdini. *Influence: The Psychology of Persuasion (Revision Edition)*. Harper Business, 2007.
- [35] Cisco. Cisco context-aware service configuration guide - 7.3. Available at [http://www.cisco.com/c/en/us/td/docs/wireless/mse/3350/7-3/CAS\\_Configuration\\_Guide/Guide/CAS\\_73/msecg\\_Overview.html](http://www.cisco.com/c/en/us/td/docs/wireless/mse/3350/7-3/CAS_Configuration_Guide/Guide/CAS_73/msecg_Overview.html).
- [36] Cisco. Rogue management in a unified wireless network - v7.4. Available at [http://www.cisco.com/c/en/us/td/docs/wireless/technology/roguedetection\\_deploy/Rogue\\_Detection.html](http://www.cisco.com/c/en/us/td/docs/wireless/technology/roguedetection_deploy/Rogue_Detection.html).
- [37] Cisco. The future of hotspots: Making wi-fi as secure and easy to use as cellular, 2012. Available at [http://www.cisco.com/en/US/solutions/collateral/ns341/ns524/ns673/white\\_paper\\_c11-649337.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns524/ns673/white_paper_c11-649337.html).
- [38] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT press, 1999.
- [39] European Commission. Proposal for a Regulation of the European parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation. See [http://ec.europa.eu/justice/data-protection/...](http://ec.europa.eu/justice/data-protection/), 0011, 2012.
- [40] G. Conti, M. Ahamad, and J. Stasko. Attacking information visualization system usability overloading and deceiving the human. In *Proc. of SOUPS 2005*, pages 89–100. ACM, 2005.

- [41] MITRE Corporation. CAPEC - Common Attack Pattern Enumeration and Classification, 2014. Available at <https://capec.mitre.org/>.
- [42] L. F. Cranor. A Framework for Reasoning About the Human in the Loop. In *Proc. of 1st Conf. on Usability, Psychology, and Security*, pages 1–15. USENIX Association, 2008.
- [43] L. F. Cranor and S. Garfinkel. *Security and Usability: Design Secure Systems that People can use*. O’Reilly Media, 2005.
- [44] M. J. C. Crump and T. M. McDonnell, J. V. and Gureckis. Evaluating Amazon’s Mechanical Turk as a tool for experimental behavioral research. *PLoS one*, 8(3):e57410, January 2013.
- [45] P. Curzon, R. Rukšėnas, and A. Blandford. An approach to formal verification of human–computer interaction. *Formal Aspects of Computing*, 19(4):513–550, 2007.
- [46] F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Adaptive Socio-Technical Systems: a Requirements-driven Approach. *Requirements Engineering*, pages 1–24, 2013.
- [47] N. David, A. David, R. R. Hansen, Kim G. Larsen, A. Legay, M. Chr. Olesen, and C. W. Probst. Modelling Social-Technical Attacks with Timed Automata. *Mist*, pages 21–28, 2015.
- [48] T. P Davis Jr. Barrier analysis facilitators guide: a tool for improving behavior change communication in child survival and community development programs. Technical report, Food for the Hungry, 2004.
- [49] T. Denning, A. Lerner, A. Shostack, and T. Kohno. Control-Alt-Hack: The Design and Evaluation of a Card Game for Computer Security Awareness and Education. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS ’13*, pages 915–928, New York, NY, USA, 2013. ACM.
- [50] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’06*, pages 581–590, New York, NY, USA, 2006. ACM.
- [51] T. Dimkov, W. Pieters, and P. H. Hartel. Portunes: Representing Attack Scenarios Spanning through the Physical, Digital and Social Domain. In A. Armando and G. Lowe, editors, *Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security - Joint Workshop, ARSPA-WITS 2010, Paphos, Cyprus, March 27-28, 2010. Revised Selected Papers*, volume 6186 of *LNCS*, pages 112–129. Springer, 2011.
- [52] D. Dolev and A. Yao. On the security of public-key protocols. *IEEE Transaction on Information Theory*, 29(2):198–208, 1983.

- 
- [53] J. Downs, A. Acquisti, and D. Barbagallo. Predictors of risky decisions: Improving judgment and decision making based on evidence from phishing attack. In *Neuroeconomics, Judgment, and Decision Making*. Psychology Press, 1 edition, 2015.
- [54] J. S. Downs, M. B. Holbrook, and L. F. Cranor. Decision strategies and susceptibility to phishing. In *Proceedings of the second symposium on Usable privacy and security*, pages 79–90. ACM, 2006.
- [55] Borgida E. and Nisbett R. E. The Differential Impact of Abstract vs. Concrete Information on Decisions. *J. of Applied Social Psychology*, pages 258–271, 1977.
- [56] C. Ellison. Ceremony Design and Analysis. Cryptology ePrint Archive, Report 2007/399, 2007.
- [57] ENISA. Detect , SHARE , Protect Solutions for Improving Threat Data Exchange among CERTs. Technical Report October, 2013.
- [58] S. Fahl, M. Harbach, T. Muders, M. Smith, L. Baumgärtner, and B. Freisleben. Why eve and mallory love android: an analysis of android SSL (in)security. In *Proc. of ACM CCS'12*, pages 50–61, New York, NY, USA, 2012. ACM.
- [59] L. Falk, A. Prakash, and K. Borders. Analyzing websites for user-visible security design flaws. In *Proceedings of SOUPS 2008*, pages 117–126, New York, NY, USA, 2008. ACM.
- [60] Federal Aviation Administration. Code of federal regulations - 14 cfr 23.777 - cockpit controls., 2002.
- [61] A. P. Felt, A. Ainslie, R. W. Reeder, S. Consolvo, S. Thyagaraja, A. Bettis, H. Harris, and J. Grimes. Improving SSL warnings: Comprehension and adherence. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pages 2893–2902, New York, NY, USA, 2015. ACM.
- [62] A. Ferreira, L. Coventry, and G. Lenzini. Principles of persuasion in social engineering and their use in phishing. In *17th Int. Conf. on Human Computer Interaction.*, page (in press), 2015.
- [63] A. Ferreira, R. Giustolisi, JL. Huynen, V. Koenig, and G. Lenzini. **Studies in Socio-technical Security Analysis: Authentication of Identities with TLS Certificates**. In *TrustCom/ISPA/IUCC*, pages 1553–1558. IEEE Computer Society, 2013.
- [64] A. Ferreira, R. Giustolisi, JL. Huynen, and G. Lenzini. **On Tools for Socio-Technical Security Analysis**. Grande Region Security and Reliability Day, 2013.

- [65] A. Ferreira, J.L. Huynen, V. Koenig, and G. Lenzini. **A Conceptual Framework to Study Socio-Technical Security**. In *HCI (24)*, volume 8533 of *Lecture Notes in Computer Science*, pages 318–329. Springer, 2014.
- [66] A. Ferreira, J.L. Huynen, V. Koenig, and G. Lenzini. **Socio-technical Security Analysis of Wireless Hotspots**. In *HCI (24)*, volume 8533 of *Lecture Notes in Computer Science*, pages 306–317. Springer, 2014.
- [67] A. Ferreira, J.L. Huynen, V. Koenig, and G. Lenzini. **In Cyber-Space No One Can Hear You S-CREAM - A Root Cause Analysis for Socio-Technical Security**. In *STM*, volume 9331 of *Lecture Notes in Computer Science*, pages 255–264. Springer, 2015.
- [68] A. Ferreira, J.L. Huynen, V. Koenig, G. Lenzini, and S. Rivas. **Socio-Technical Study on the Effect of Trust and Context When Choosing WiFi Names**. In *STM*, volume 8203 of *Lecture Notes in Computer Science*, pages 131–143. Springer, 2013.
- [69] A. Ferreira, J.L. Huynen, V. Koenig, G. Lenzini, and S. Rivas. **Do Graphical Cues Effectively Inform Users? - A Socio-Technical Security Study in Accessing Wifi Networks**. In *HCI (22)*, volume 9190 of *Lecture Notes in Computer Science*, pages 323–334. Springer, 2015. [Best Paper Award].
- [70] A. Ferreira and G. Lenzini. An analysis of social engineering principles in effective phishing. In *Socio-Technical Aspects in Security and Trust (STAST), 2015 Workshop on*, pages 9–16, July 2015.
- [71] J. Finch. The Vignette Technique in Survey Research. *Sociology*, 21(1):105–114, 1987.
- [72] I. Fléchaïs. *Designing secure and usable systems*. PhD thesis, University College London, 2005.
- [73] International Organization for Standardization. *Information Technology; Security Techniques; Information Security Management Guidelines for Telecommunications Organizations Based on ISO: Technologies de L'information: Techniques de Sécurité: Lignes Directrices Pour Les Organismes de Télécommunications Sur la Base de L'ISO/CEI 27002*. International Organization for Standardization, 2009.
- [74] D. Gambetta. *Trust: Making and Breaking Cooperative Relations*. Blackwell Pub, 1990.
- [75] D. Gambetta. Can We Trust Trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213–237. Basil Blackwell, 2000.
- [76] G. Gavai, K. Sricharan, D. Gunning, R. Rolleston, J. Hanley, and M. Singhal. Detecting insider threat from enterprise social and online activity data. In *Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats*, MIST '15, pages 13–20, New York, NY, USA, 2015. ACM.

- 
- [77] M. Georgiev, S. Iyengar, S. Jana, Rishita A., D. Boneh, and V. Shmatikov. The most dangerous code in the world: validating SSL certificates in non-browser software. In *Proc. of ACM CCS'12*, pages 38–49, New York, NY, USA, 2012.
- [78] P. Godfrey-Smith. *Theory and Reality: An Introduction to the Philosophy of Science*. Science and Its Conceptual Foundations. Univ. of Chicago Press, 2009.
- [79] D. Gollmann. *Computer Security*. Wiley, 2011.
- [80] Google. AngularJS. Available at <https://angularjs.org/>.
- [81] D. Gragg. A Multi-Level Defense Against Social Engineering. Technical report, SANS Institute - InfoSec Reading Room, 2003.
- [82] P. Grice. *Studies in the Way of Words*. Harvard University Press, 1989.
- [83] K. H. Guo and Y. Yuan. The effects of multilevel sanctions on information security violations: A mediating model. *Information & management*, 49(6):320–326, 2012.
- [84] J. T. Hallinan. *Why We Make Mistakes: How We Look Without Seeing, Forget Things in Seconds, and Are All Pretty Sure We Are Way Above Average*. Broadway Books, 2009.
- [85] D. Harley. Re-floating the titanic: Dealing with social engineering attacks. *London: EICAR*, page 13, 1998.
- [86] R. Heartfield and G. Loukas. A Taxonomy of Attacks and a Survey of Defence Mechanisms for Semantic Social Engineering Attacks. *ACM Computing Surveys*, 48(3):1–39, 2015.
- [87] C. Herley. So Long, And No Thanks for the Externalities: the Rational Rejection of Security Advice by Users. In *Proceedings of the 2009 workshop on New security paradigms workshop*, NSPW '09, pages 133–144, New York, NY, USA, 2009. ACM.
- [88] C. Herley. Why do Nigerian Scammers Say They are From Nigeria? *Workshop on the Economics of Information Security (WEIS)*, 2012.
- [89] R. J. Heuer. *Psychology of Intelligence Analysis*, 1999.
- [90] A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall International, 1985.
- [91] E. Hollnagel. *Cognitive reliability and error analysis method CREAM*. Elsevier, Oxford New York, 1998.
- [92] A. Hovav and F. F. Putri. This is my device! why should I follow your rules? employees' compliance with BYOD security policy. *Pervasive and Mobile Computing*, 2016.

- [93] R. Huber. Distributed security alerting, March 2016. Available at <https://slack.engineering/distributed-security-alerting-c89414c992d6>.
- [94] J. Hunker and C.W. Probst. Insiders and insider threats an overview of definitions and mitigation techniques. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2(1):4–27, 2011.
- [95] JL. Huynen. S-CREAM Assistant, a tool to support S-CREAM analyses., 2016. Available at <https://github.com/gallypette/SCREAM>.
- [96] IBM. IBM Security Services 2014 Cyber Security Intelligence Index., 2014. Available at [http://media.scmagazine.com/documents/82/ibm\\_cyber\\_security\\_intelligenc\\_20450.pdf](http://media.scmagazine.com/documents/82/ibm_cyber_security_intelligenc_20450.pdf).
- [97] 802.11u-2011–amendment 9: Interworking with external networks.
- [98] International Civil Aviation Organization (ICAO). *ICAO ADREP 2000 taxonomy*, 2000.
- [99] Isaca. *COBIT 5 Framework*. Isaca, 2012.
- [100] K. Ishikawa. *Introduction to Quality Control*. Springer, 2012.
- [101] K. Ivaturi and L. Janczewski. A taxonomy for social engineering attacks. In *International Conference on Information Resources Management, Centre for Information Technology, Organizations, and People (June 2011)*, 2011.
- [102] D. Jackson. Towards a theory of conceptual design for software. In *Onward! Essays*, 2015.
- [103] L.J. Janczewski and F. Lingyan. Social engineering-based attacks: Model and New Zealand perspective. In *Proc. of IMCSIT 2010*, pages 847–853, 2010.
- [104] D. Jeske, L. Coventry, and P. Briggs. Decision justifications for wireless network selection. In *Socio-Technical Aspects in Security and Trust (STAST), 2014 Workshop on*, pages 1–7, July 2014.
- [105] C. Johnson, L. Badger, and D. Waltermire. NIST special publication 800-150 (draft) guide to cyber threat information sharing. Technical report, The National Institute of Standards and Technology (NIST), October 2014.
- [106] M. Jordan and H. Gouday. The signs, and semiotics of the successful semantic attack. In *14th Annual EICAR Conference*, pages 344–364, 2005.
- [107] A. Jøsang, I. G. Pedersen, and D. Povey. PKI seeks a trusting relationship. In *Proc. of ACISP 2000, Brisbane, Australia*, 2000.
- [108] A. Jøsang, K. A. Varmedal, C. Rosenberger, and R. Kumar. Service provider authentication assurance. In *Proc. of PST '12*, pages 203–210. IEEE Computer Society, 2012.
- [109] Js-data Development Team. Js-data. Available at <http://www.js-data.io/>.



- 
- [110] R. Kainda, I. Fléchais, and A. W. Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *Proc. of the 5th Symposium on Usable Privacy and Security, SOUPS '09*, pages 11:1–11:12, New York, NY, USA, 2009. ACM.
- [111] R. Kainda, I. Fléchais, and A. W. Roscoe. Security and Usability: Analysis and Evaluation. *2010 International Conference on Availability, Reliability and Security*, pages 275–282, 2010.
- [112] I. Kirlappos, A. Beauteument, and M. A. Sasse. “Comply or Die” Is Dead: Long Live Security-Aware Principal Agents, pages 70–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [113] I. Kirlappos, S. Parkin, and M. A. Sasse. Learning from “shadow security:” why understanding non-compliant behaviors provides the basis for effective security. In *Proceedings 2014 Workshop on Usable Security*. Internet Society, 2014.
- [114] P. Klasnja, S. Consolvo, J. Jung, B. M. Greenstein, L. LeGrand, P. Powledge, and D. Wetherall. “When I am on Wi-Fi, I am fearless”: privacy concerns & practices in everyday Wi-Fi use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pages 1993–2002, New York, NY, USA, 2009. ACM.
- [115] R. Koppel, S. W. Smith, J. Blythe, and V. Kothari. Workarounds to computer access in healthcare organizations: You want my password or a dead patient? In *Driving Quality in Informatics: Fulfilling the Promise, ITCH 2015, Victoria, BC, Canada, February 26 - March 1, 2015*, pages 215–220, 2015.
- [116] K.P.K. Kumar and G. Geethakumari. Analysis of semantic attacks in online social networks. *Communications in Computer and Information Science*, 420 CCIS:45–56, 2014. cited By 0.
- [117] P. Kumaraguru, S. Sheng, A. Acquisti, L. F. Cranor, and J. Hong. Teaching Johnny not to Fall for Phish. *ACM Trans. Internet Technol.*, 10(2):1–31, 2010.
- [118] N. Kuntze, C. Rudolph, B. Brisbois, M. Boggess, B. Endicott-Popovsky, and S. Leivesley. Safety vs. security: Why do people die despite good safety? In *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, pages 1–13, April 2015.
- [119] A. Langley. Living with HTTPS, July 2012.
- [120] L. Laribee. *Development of methodical social engineering taxonomy project*. PhD thesis, Monterey, California. Naval Postgraduate School, 2006.
- [121] J. Lazar, J.H. Feng, and H. Hochheiser. *Research Methods in Human-Computer Interaction*. John Wiley & Sons, 2010.

- [122] E. Lee. *Homeland Security and Private Sector Business: Corporations' Role in Critical Infrastructure Protection, Second Edition*. CRC Press, 2014.
- [123] M. G. Lee. Securing the human to protect the system: Human factors in cyber security. In *System Safety, incorporating the Cyber Security Conference 2012, 7th IET International Conference on*, pages 1–5, Oct 2012.
- [124] E. L. Lehmann. 'student' and small-sample theory. *STATISTICAL SCIENCE*, 14:418–426, 1999.
- [125] Periscope Film Llc. *B-17 Bomber Pilot's Flight Operating Manual*. lulu.com, 2013.
- [126] G. B. Magklaras and S. M. Furnell. Insider Threat Prediction Tool: Evaluating the probability of IT misuse. *Computers & Security*, 21(1):62 – 73, 2001.
- [127] Mandiant. Sophisticated indicators for the modern threat landscape: An introduction to OpenIOC. Available at [http://openioc.org/resources/An\\_Introduction\\_to\\_OpenIOC.pdf](http://openioc.org/resources/An_Introduction_to_OpenIOC.pdf).
- [128] S. Mauw and M. Oostdijk. Foundations of attack trees. In D. Won and S. Kim, editors, *Proc. of the 8th Int. Conf. on Information Security and Cryptology (ICISC 2005), December 1-2, 2005, Seoul, Korea*, volume 3935 of *LNCS*, pages 186–198. Springer, 2006.
- [129] P. McCullagh. *Generalized linear models*. Chapman and Hall, London New York, 1989.
- [130] MISP Development Team. Malware Information Sharing Platform, 2015. Available at <http://www.misp-project.org/>.
- [131] K. Mitnick and W. L. Simon. *the Art of Deception*. Wiley Publishing Inc., 2002.
- [132] D. Miyamoto and T. Takahashi. Toward automated reduction of human errors based on cognitive analysis. In *Proceedings of the 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS '13*, pages 820–825, Washington, DC, USA, 2013. IEEE Computer Society.
- [133] C. F. Mohd Foozy, R. Ahmad, M. A. Faizal, R. Yusof, and M. M. Zaki. Generic taxonomy of social engineering attack. Technical Report 191, University Teknikal Malaysia Melaka, 2011.
- [134] F. Mouton, L. Leenen, M. M. Malan, and H. S. Venter. *ICT and Society: 11th IFIP TC 9 International Conference on Human Choice and Computers, HCC11 2014, Turku, Finland, July 30 – August 1, 2014. Proceedings*, chapter Towards an Ontological Model Defining the Social Engineering Domain, pages 266–279. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

- 
- [135] National Security Agency. Defense in depth: A practical strategy for achieving information assurance in today's highly networked environments.
- [136] Jakob Nielsen, editor. *Designing User Interfaces for International Use*. Elsevier Science Publishers Ltd., Essex, UK, 1990.
- [137] European Council of the European Union. EU steps up cybersecurity: member states approve agreement, 2015.
- [138] G. L. Orgill, G. W. Romney, M. G. Bailey, and P. M. Orgill. The urgency for effective user privacy-education to counter social engineering attacks on secure computer systems. In *Proc. of the 5th conference on Information technology education, CITC5 '04*, pages 177–181, New York, NY, USA, 2004. ACM.
- [139] E. Paja, F. Dalpiaz, and P. Giorgini. Sts-tool: Security requirements engineering for socio-technical systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8431:65–96, 2014.
- [140] E. Paja, M. Poggianella, F. Dalpiaz, P. Roberti, and P. Giorgini. Security requirements engineering with sts-tool. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8900:95–109, 2014.
- [141] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.
- [142] S. Parkin, A. van Moorsel, P. G. Inglesant, and M. A. Sasse. A Stealth Approach to Usable Security: Helping IT Security Managers to Identify Workable Security Solutions. In *Proc. of NSPW 2010, Sept. 21-23, 2010*, pages 33–50. ACM, 2010.
- [143] PCI SSC. Requirements and Security Assessment Procedures v3.1, 2015.
- [144] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. Available at <http://www.R-project.org>.
- [145] D. G. Rand. The promise of Mechanical Turk: How online labor markets can help theorists run behavioral experiments. *Journal of Theoretical Biology*, 299:172–179, 2012.
- [146] A. Raskin. Tabnabbing: A New Type of Phishing Attack. Available at <http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/>.
- [147] J. Reason. *Human Error*. Cambridge University Press, 1990.
- [148] E. Rescorla. HTTP Over TLS. RFC 2818, 2000.
- [149] R. Rukšėnas, P. Curzon, and A. Blandford. Modelling and Analysing Cognitive Causes of Security Breaches. *Innovation in Systems and Software Engineering*, 4(2):143–160, 2008.

- [150] S. Ruoti, J. Andersen, D. Zappala, and K. Seamons. Why Johnny Still, Still Can't Encrypt: Evaluating the Usability of a Modern PGP Client. Technical report, arXiv.org, oct 2015.
- [151] S.E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The Emperor's New Security Indicators. *IEEE Symposium on Security and Privacy (SP '07)*, pages 51–65, 2007.
- [152] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. Wiley, 2000.
- [153] B. Schneier. Semantic attacks: The third wave of network attacks. *Cryptogram Newsletter*, 14, 2000.
- [154] R. D. Serwy and E. M. Rantanen. Evaluation of a software implementation of the cognitive reliability and error analysis method (CREAM). *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 51(18):1249–1253, oct 2007.
- [155] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs. Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 373–382. ACM, 2010.
- [156] A. Sood and R. Enbody. *Targeted Cyber Attacks: Multi-staged Attacks Driven by Exploits and Malware*. Elsevier Science, 2014.
- [157] A. Sotirakopoulos, K. Hawkey, and K. Beznosov. On the challenges in usable security lab studies: lessons learned from replicating a study on SSL warnings. In *Proceedings of the Seventh Symposium on Usable Privacy and Security, SOUPS '11*, pages 3:1–3:18, New York, NY, USA, 2011. ACM.
- [158] S. Srinivas, D. Balfanz, E. Tiffany, and A. Czeskis. Universal 2nd Factor (U2F) Overview., 2015. Available at <https://fidoalliance.org/specifications/download/>.
- [159] F. Stajano and P. Wilson. Understanding Scam Victims: Seven Principles for Systems Security. *Commun. ACM*, 54(3):70–75, March 2011.
- [160] D. Stakenburg and J. Crampton. Underexposed risks of public wi-fi hotspots. *Computer Weekly.Com*, 2013.
- [161] S. Stasiukonis. Social engineering, the USB way. *Dark Reading*, 7, 2006.
- [162] D. F. Sterne. On the buzzword 'security policy'. In *Research in Security and Privacy, 1991. Proceedings., 1991 IEEE Computer Society Symposium on*, pages 219–230, May 1991.
- [163] B. Strauch. *Investigating Human Error: Incidents, Accidents, and Complex Systems*. Ashgate Pub Ltd, 2004.

- 
- [164] J. Sun, Y. Liu, J. S. Dong, and J. Pang. *PAT: Towards Flexible Verification under Fairness*, pages 709–714. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [165] J. Sunshine, S. Egelman, H. Almuhiemedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of SSL warning effectiveness. In *Proc. of USENIX'09*, 2009.
- [166] A. D. Swain. THERP. Technical report, Sandia Corp., Albuquerque, N. Mex., 1964.
- [167] A.D. Swain, U.S. Nuclear Regulatory Commission. Office of Nuclear Regulatory Research, and H.E. Guttman. *Handbook of Human Reliability Analysis With Emphasis on Nuclear Power Plant Applications - Draft Report For Interim Use and Comment*. NUREG/CR. U.S. Nuclear Regulatory Commission, 1980.
- [168] R. Tembe, Kyung Wha Hong, E. Murphy-Hill, C.B. Mayhorn, and C.M. Kelley. American and indian conceptualizations of phishing. In *Proc. of STAST 2013*, pages 37–45. IEEE, 2013.
- [169] P. Tetri and J. Vuorinen. Dissecting social engineering. *Behaviour & Information Technology*, 32(10):1014–1023, 2013.
- [170] The News York Times. An old swindle revived; The “Spanish Prisoner” and Buried Treasure Bait Again Being Offered to Unwary Americans. 1898.
- [171] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185:1124–1131, 1974.
- [172] A. Tversky and D. Kahneman. Rational Choice and The Framing of Decisions. *J. Business*, 59:251–278, 1986.
- [173] B. Ur, F. Noma, J. Bees, S. M. Segreti, R. Shay, L. Bauer, N. Christin, and L. F. Cranor. “I added ‘!’ at the end to make it secure”: Observing password creation in the lab. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 123–140, 2015.
- [174] US Bureau of Labor Statistics. Standard Occupational Classification and Coding Structure. Technical Report February, 2010.
- [175] U.S. Department of the Census. *Current Population Survey interviewing manual*, June 2013.
- [176] VCDB Development Team. VERIS Community Database. Available at <http://vcdb.org/>.
- [177] Verizon. VERIS: The Vocabulary for Event Recording and Incident Sharing. Available at <http://veriscommunity.net/>.

- [178] M. Volkamer and K. Renaud. Mental models-general introduction and review of their application to human-centred security. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8260 LNCS:255–280, 2013. cited By 2.
- [179] M. Volkamer, S. Stockhardt, S. Bartsch, and M. Kauer. Adopting the CMU/APWG anti-phishing landing page idea for germany. In *Proc. of STAST 2013*, pages 46–52. IEEE, 2013.
- [180] Y. Wang, P. G. Leon, A. Acquisti, L. F. Cranor, A. Forget, and N. Sadeh. A field trial of privacy nudges for facebook. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pages 2367–2376, 2014.
- [181] We Are Dynamo turker community. Guidelines for Academic Requesters. Available at <http://www.wearedynamo.org/>.
- [182] D. Weerasinghe, V. Rakocevic, and M. Rajarajan. Security framework for mobile banking. In *Proc. of 8th MoMM 2010*, pages 421–424. ACM, 2010.
- [183] R. West. The Psychology of Security. *Communication of the ACM*, 51(4):34–38, April 2008.
- [184] A. Whitten. Why Johnny can’t encrypt: A usability evaluation of PGP 5.0. *Proceedings of the 8th USENIX Security*, 1999.
- [185] B. Whitworth. Socio-technical systems. *Encyclopedia of human computer interaction*, pages 533–541, 2006.
- [186] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.
- [187] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, page 601, 2006.
- [188] yubico AB. Yubikey 4 and yubikey 4 nano. Available at <https://www.yubico.com/products/yubikey-hardware/yubikey4/>.
- [189] yubico AB. Yubikey security evaluation: Discussion of security properties and best practices., 2012. Available at <https://www.yubico.com/wp-content/uploads/2012/10/Security-Evaluation-v2.0.1.pdf>.
- [190] yubico AB. The yubikey manual: Usage, configuration and introduction of basic concepts., 2015. Available at [https://www.yubico.com/wp-content/uploads/2015/03/YubiKeyManual\\_v3.4.pdf](https://www.yubico.com/wp-content/uploads/2015/03/YubiKeyManual_v3.4.pdf).
- [191] yubico AB. OTP vs. U2F: Strong to stronger., 2016. Available at <https://www.yubico.com/2016/02/otp-vs-u2f-strong-to-stronger/>.

- [192] Y. Zhang, F. Monrose, and M. K. Reiter. The security of modern password expiration: An algorithmic framework and empirical analysis. In *Proc. of the 17th ACM Conf. on Computer and Communication Security (CCS'10)*, October 4-8, 2010, Chicago, IL, USA, pages 176–186. ACM, 2010.