FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Usable Security and Privacy Challenges with Disruptive Technologies

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin der Technischen Wissenschaften

eingereicht von

Dipl.-Ing. Katharina Krombholz-Reindl, BSc.
Matrikelnummer 0508215

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Edgar Weippl

Diese Dissertation haben begutachtet:

Peter Purgathofer

Stefanie Rinderle-Ma

Wien, 25. August 2016

Katharina Krombholz-Reindl

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Usable Security and Privacy Challenges with Disruptive Technologies

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktorin der Technischen Wissenschaften

by

Dipl.-Ing. Katharina Krombholz-Reindl, BSc.
Registration Number 0508215

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Edgar Weippl

The dissertation has been reviewed by:

Peter Purgathofer	Stefanie Rinderle-Ma

Vienna, 25th August, 2016

Katharina Krombholz-Reindl

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Katharina Krombholz-Reindl, BSc.
Obere Amtshausgasse 38/13, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 25. August 2016

Katharina Krombholz-Reindl

Acknowledgements

I am gratefully indebted to my advisor Edgar Weippl for making this thesis possible. The door to Edgar's office was always open to discuss my research and he allowed me the freedom to pursue my own research ideas and projects.

I would also like to thank SBA Research for providing me a great environment to develop, discuss and publish my work. I am deeply grateful to my fellows at SBA for all the insightful discussions, fruitful collaborations and especially the fun we had throughout the recent years. In particular, I would like to thank Adrian Dabrowski, Martin Schmiedecker, Martina Lindorfer, Wilfried Mayer and Aljosha Judmayer. I highly value the relaxed but productive work environment, the unconditional support we provided each other and last but not least our friendship.

During my studies, I had the distinct pleasure to collaborate with great researchers from other institutions. My sincere thanks go to Thorsten Holz who provided me the opportunity to spend some time at Ruhr-University Bochum to collaborate with his group. I very much enjoyed working with Thorsten and his group and it was a great learning experience. Furthermore I am deeply grateful to Matthew Smith from the University of Bonn for his support and the valuable feedback I received whenever I needed it. Matthew's support especially helped me to gain confidence in my own ideas. Furthermore, I would like to thank Peter Purgathofer for his inspiring lectures that I have enjoyed since I started my bachelor studies. I highly appreciate discussing my ideas with him in the course of lectures and also beyond. I am also grateful to Dieter Merkl who supervised my master thesis and then encouraged me to start a PhD. Without Dieter's support I would probably not have considered starting a PhD.

I must express my very profound gratitude to my family and partner, in particular my parents Irene Krombholz and Martin Krombholz, my brothers Philipp and Thomas and my sister Barbara for providing me with love, unfailing support and continuous encouragement. I am also deeply thankful to my partner Andreas Hubmer for his love and support and the great time we had since we met each other in one of the first computer science lectures of our bachelor studies. Without the constant support from my beloved ones throughout my life this thesis would not have been possible.

My thanks also go to all my friends for their patience and moral support in recent years, in particular Bine, Caro, Andi K., Anka, Karin, Lända, Gergö, Verena, Chrizzy and Julia.

Kurzfassung

"Security is only as good as its weakest link, and people are the weakest link in the chain." (Bruce Schneier)

Im täglichen Leben interagieren BenutzerInnen mit einer Vielzahl von technischen Geräten, welche kontinuierlich Daten über ihre BenutzerInnen sowie deren Umgebung sammeln und dann über das Internet übertragen. Im digitalen Zeitalter werden neue Technologien schnell in den Alltag integriert, was dazu führt, dass eine immer größere Anzahl an Geräten online ist und Daten austauscht. Dieses neue Paradigma impliziert jedoch neue Herausforderungen, da die BenutzerInnen mit immer komplexeren Informationsverarbeitungskonzepten konfrontiert werden. Auch die Forschung beschäftigt sich seit einigen Jahren mit diesem Aspekt der Informationssicherheit.

EndanwenderInnen empfinden die Nutzung von Sicherheitssystemen häufig als schwierig, da diese oftmals nicht unter der Berücksichtigung der Ansprüche von EndanwenderInnen gestaltet wurden. Daher sind diese Systeme anfällig für Angriffe von außen. Ausserdem besteht das Risiko, dass EndanwenderInnen unwissentlich sensible Daten mit Dritten teilen. Aus diesen Gründen ist eine Integration von Aspekten der Mensch-Maschine Interaktion in die Sicherheitsforschung notwendig. Der interdisziplinäre Forschungsbereich, der sich mit dieser Thematik beschäftigt, wird *Usable Security* genannt.

Das übergeordnete Ziel dieser Arbeit ist es einen Beitrag zur besseren Benutzbarkeit von Sicherheitssystemen zu leisten. Dies geschieht zum einen mittels Benutzerstudien um die Interaktionen der Menschen mit dem System besser zu verstehen und zum anderen um mittels neuer Designkonzepte den Ansprüchen der Nutzer besser gerecht zu werden.

Im Rahmen dieser Arbeit haben wir uns mit unterschiedlichen Herausforderung im Bereich der Usable Security auseinandergesetzt. Zuerst haben wir Social Engineering Angriffe systematisiert und mittels Machine Learning Onlineplattformen als Untergrundmärkte, auf denen gestohlene Daten gehandelt werden, klassifiziert. Anschließend haben wir Phishingattacken über QR Codes untersucht, sowie benutzerzentrierte Gegenmaßnahmen entwickelt und evaluiert. Des Weiteren haben wir explorativ Designansätze zum Schutz der Privatsphäre gegenüber in Wearables eingebauten Kameras im öffentlichen Raum erforscht. Mittels qualitativer Interviews konnten wir nutzerfreundliche Formfaktoren für ein Tool zur Mediation zwischen dem Träger eines Wearables und zufällig fotografierten Personen in

der Umgebung identifizieren. Unsere Ergebnisse zeigen, dass ein eigens dafür konzipiertes Gerät, welches mit Tasten einfach bedienbar ist, dafür am besten geeignet wäre. Um die Datensicherheit auf Smartphones zu verbessern haben wir ein drucksensibles Verfahren namens Force-PINs zur Benutzerauthentifizierung entwickelt und im Rahmen zweier Benutzerstudien gezeigt, dass dieses Verfahren mit nur minimaler Verschlechterung der Benutzbarkeit einhergeht und sicherere PINs ermöglicht. Darüber hinaus haben wir zwei große Benutzerstudien zu Kryptographischen Anwendungen durchgeführt. Zuerst haben wir uns der Kryptowährung Bitcoin gewidmet und eine großangelegte Onlinebefragung mit 990 Bitcoin BenutzerInnen durchgeführt, um Herausforderungen bezüglich Sicherheit und Privatsphäre aus Nutzersicht zu erforschen. Unsere Ergebnisse zeigen, dass selbst technikaffine BenutzerInnen Schwierigkeiten beim Schutz ihres digitalen Vermögens haben. Des weiteren haben wir eine Nutzerstudie mit gut ausgebildeten TeilnehmerInnen durchgeführt um Herausforderung bei der Umsetzung von HTTPS zum Schutz von Webseiten aus Nutzerperspektive zu erforschen. Unsere Ergebnisse zeigen, dass viele schlechte Konfigurationen durch missverständliche Benutzerschnittstellen während des Deployment Prozesses entstehen.

Die Ergebnisse unserer Forschung in verschiedenen Anwendungsfeldern haben zukünftige Herausforderungen und Fragestellungen ans Licht gebracht. Des Weiteren konnten wir benutzerzentrierte Designs für eine bessere Benutzerbarkeit von Sicherheitsmechanismen entwickeln und den Mehrwehrt im Rahmen von Nutzerstudien zeigen.

Abstract

"Security is only as good as its weakest link, and people are the weakest link in the chain." (Bruce Schneier)

In the current age, disruptive technologies are proliferating rapidly and a plethora of devices is interconnected and exchanges data. This always-online paradigm poses significant challenges to their users as the underlying information-sharing models are difficult to understand. Hence, managing security and privacy has become increasingly complex for users. This complexity is more and more acknowledged and research has started to address human aspects of information security.

End-users often struggle with security systems that are too difficult to use and not designed to fulfil the users' needs. As a result, they are susceptible to a variety of attacks or accidentally disclose sensitive information without being aware of it. This highlights the need for an integration of human-computer interaction aspects in security research. This interdisciplinary field which is also referred to as *usable security* has become necessary and is currently an emerging field of research.

The goal of this work is to contribute to making security and privacy technology more user-friendly by understanding the users through user studies and by providing new concepts and designs that fulfil the users' needs.

Throughout this thesis, we focused on usable security challenges around disruptive technologies. First, we systematized social engineering attack vectors and used machine learning to detect underground marketplaces where stolen sensitive data is traded. Then, we studied QR code-based phishing attacks and proposed and evaluated user-centric mitigation strategies. Moreover, we explored design directions for the design of future privacy-mediating technologies to support informed consent between users of wearable cameras in public places. Through qualitative interviews, we determined form factors for future designs and found that the participants preferred a tangible and decentralized device with a simple button to push. Furthermore, we proposed an enhanced PIN scheme called *force-PINs* and showed that our approach supports users in selecting stronger PINs with only minimal task overhead compared to digit-only PINs. We furthermore conducted user studies to research security and privacy-related challenges of crypto applications such as Bitcoin and TLS. Our large-scale study with Bitcoin users revealed that even

experienced users often lose their keys and insufficiently backup their digital assets. The results of a lab study to study usability challenges in the HTTPS deployment process suggest that administrators are confronted with poor usability which results in weak configurations.

Our findings in various fields of application revealed future challenges for the design of usable security and privacy technology based on user studies. Also, we presented user-centric security schemes and showed that our approaches improve security with a reasonable task overhead.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The Snowden revelations on large-scale government surveillance and the plethora of everyday devices connected to the Internet have sparked a wave of public concern regarding security and privacy in end user technology. Due to the subsequent debate on online privacy and data security in mainstream media, user have become aware of the value in their personal data. Since then, web searches on privacy-enhancing technologies such as "encryption", "PGP" and "TOR" have increased according to Preibusch et al. [152]. Also, Facebook reported diminishing trust among their users. However, they also reported that there was no impact on the frequency of use of its social network [152].

Usage statistics on mobile messaging also show that end-to-end encrypted messaging apps are still niche products despite the increased awareness for privacy among the population. One of the main reasons why these tools still lack from large-scale adoption is that most of them are difficult to use for non-expert users. Furthermore, security and privacy technology is often perceived as disruptive as it requires users to perform additional tasks and therefore distracts them from their main task when operating a system. Also, the paradigm shift to a world where devices are continuously collecting, storing and sharing data of their users and surroundings makes it difficult for users to understand what information they share with whom and how to protect their data from unwanted access.

These factors impact distinct challenges to the usability of security and privacy technology. Therefore, an integration of human-computer-interaction aspects into security research is necessary to make security tools more user-friendly and to prevent systems from being compromised.

The field of usable security and privacy is an emerging field in the area of information security. Traditional security research aimed at making computers, networks and embed-

ded systems trustworthy. While such secure systems offer significant benefits in theory, they have been shown to be too difficult for people to use in practice. As a result, end users move back to less secure solutions or find ways to circumvent their interactions with the security tool. Probably the most striking example is e-mail encryption which is rarely being adopted by non-expert users. Novel paradigms such as the Internet of Things rely on complex information-sharing models that users are often not aware of. Therefore systems must be designed to be inherently usable secure beyond predictable attack scenarios without requiring the users deeper understanding of the underlying fundamentals. They are furthermore required to provide security independently from careless or unmotivated users.

1.2 Goals of this Work

The goal of this thesis is to address usable security and privacy challenges of applications that reflect disruptive novel paradigms. To research these technologies, we aim at bridging the gap between security, privacy and usability/UX by combining research methods from these disciplines to get an interdisciplinary view on both security and human aspects.

In particular, we focus on the following topics: (1) user-centric attack vectors and underground marketplaces, (2) QR code security, (3) privacy in the age of wearable computing, (4) user authentication on mobile devices and (5) usability of cryptographic applications. To address usable security and privacy challenges around these scenarios, we aim at understanding users and how they interact with the technology on the one hand, and on the other hand we propose novel user-centric designs.

Regarding user-centric attack vectors, we present a taxonomy of state of the art social engineering attacks in the Internet ecosystem, and an automated tool to determine whether an Internet source is an underground marketplace where through social engineering acquired sensitive information is traded. The goals were to categorize different attack scenarios with respect to modern communication channels and to understand how socially engineered information is then traded on online platforms.

The goal of our research on QR code security was to assist users in deciding whether a QR code they are about to scan comes from a trustworthy source or has potentially harmful code or information encoded. To address this goal, we started with an evaluation of QR code reader apps and a social engineering experiment. Then, we proposed and evaluated design guidelines.

In the domain of wearable computing, we evaluated the state of the art in privacy-mediating- and privacy-enhancing technologies and present an explorative qualitative user study to determine form factors for future privacy-mediating technology. The goals of this research were to understand the state of the art regarding currently available technology that prevents unconsented recording of bystanders, and to determine form factors for a user-centric design of a privacy-mediating technology in the age of wearable computing.

Regarding user authentication on mobile devices, we present an enhanced knowledge-based authentication scheme based on pressure-sensitive digit input. As smartphones contain potentially sensitive personal data, the protection of the latter is necessary to prevent unwanted access. As authentication schemes used in most modern smartphones are rather weak, our goal was to improve the security of an already existing scheme with minimal impact on task overhead.

In the area of usable cryptography, we focused on two specific application scenarios, namely the crypto-currency Bitcoin and Transport Layer Security (TLS). The main goal was to identify usability concepts that weaken security. Regarding Bitcoin, we performed a user study to understand how users interact with the crypto-currency and to research their experiences with security, privacy and anonymity. Regarding TLS, we focussed on expert users to measure the impact of poor usability on the security of the resulting server configuration.

1.3 Methodology

To collect evidence for the research work presented in this thesis, we followed an interdisciplinary approach and combined methods from security research and human-computer interaction. Most our results follow from a mixed-method approach where data is acquired through multiple methods and then triangulated to ensure validity. Every topic was researched in three phases: We started with a literature review and then performed an analysis of how users interact with the system under investigation to determine design challenges. Based on our findings, we proposed implications for future designs or presented and evaluated completely new designs.

In particular, we used the following research methods:

- **Comprehensive literature review** on the current state of the art in usable security research and **systematization of knowledge** to evaluate, systematize, and contextualize existing knowledge.
- **Prototype implementations** to evaluate conceptual approaches in user studies, i.e. a novel authentication scheme for smartphones.
- **Cognitive walkthroughs** to evaluate already existing technology with respect to human-centric concerns. We extended our cognitive walkthroughs with techniques such as HTTP(S) intersection to detect vulnerabilities.
- **Machine learning** to classify communication channels.
- **User studies** to evaluate security approaches towards human-centric concerns. We used the following types of user studies:

- **Online surveys** to collect large-scale, mostly quantitative data through single/multiple choice questions and Likert scales. In addition, we used open-ended questions to collect qualitative data.
- **Semi-structured interviews** to gather qualitative data for a deeper understanding on socially based phenomena in usable security that cannot be easily quantified or experimentally manipulated or, for that matter, ethically researched with experiments. We used these methods mainly to explore new technology paradigms with respect to the context, e.g. wearables and privacy in public spaces and Bitcoin.
- **Lab studies** following both within- and between-subjects designs to experimentally evaluate prototype implementations. We evaluated our approaches both against usability metrics and against specific threat models.
- To gather evidence and insights to how our approaches were deployed in the wild, we conducted **field studies** over a period of time via an experience sampling method on a mobile device.

We started our research with looking at user-centric attack vectors and the implications of leakage of private data in underground marketplaces. To do so, we conducted an extensive literature review and systematized social engineering attack vectors with respect to communication channels and socio-technical aspects. Also, we identified underground marketplaces where stolen and leaked personal data (e.g., credit card information) was traded by attackers. In order to facilitate the detection of such underground marketplaces, we used machine learning to train a classifier and to automate the process of deciding whether a communication channel is used as an underground marketplace.

For our research on QR code security, we conducted a literature review to identify major research challenges and then performed cognitive walkthroughs and HTTP(S) interception to evaluate the twelve most frequently downloaded QR code reader apps regarding anti-phishing and other security features. We also conducted a phishing experiment in the field, where we deployed QR codes in public places with a link to a survey encoded.

For our research on privacy challenges for bystanders of wearable cameras, we systematized existing privacy-mediating technology and found that many of these technologies are only applicable in narrow, pre-defined scenarios and often lack of usability. Based on these findings, we conducted qualitative semi-structured interviews in the field with 20 participants. The data was analyzed using an iterative coding approach.

Furthermore, we proposed an enhanced PIN-based authentication scheme and implemented a prototype iOS app which was then used in a user study and, in a slightly modified version, for a two-week long field study. The lab study with 50 participants followed a within-subjects design to evaluate three conditions. After being exposed to all conditions, the participants filled out a questionnaire with both closed- and open-ended questions. The field study was conducted with 10 participants who used the

authentication scheme several times a day over two weeks. The data was collected via comprehensive logging and in-situ qualitative comments.

For our research on user experiences with Bitcoin security, privacy, anonymity and usability we conducted a large-scale online survey with 990 participants consisting of both closed-ended questions to gather quantitative data and a set of open-ended questions to collected qualitative data. To get a deeper understanding on the large-scale phenomena we determined, we conducted additional semi-structured interviews with a subset of 10 participants from the online survey.

Our research on TLS usability focused on expert-users instead of non-expert users. We conducted a lab study with 30 participants to identify usability flaws in the TLS deployment workflow. We used a think-aloud method to collect qualitative data and applied a logging mechanism to implement experience sampling. Additionally, we conducted semi-structured interviews with domain experts.

1.4 Scientific Contribution

In the following, we provide an overview of the scientific contributions covered by the peer-reviewed papers referred to in this thesis.

1.4.1 User-Centric Attack Vectors and Underground Marketplaces

Regarding user-centric attack vectors, we performed a comprehensive literature analysis and systematized state of the art social engineering attacks against knowledge workers.The goal was to provide an overview of the most commonly occurring social engineering attacks with respect to socio-technical aspects. As stolen data is often traded in underground marketplaces, we investigated online communication channels and built a text classifier to determine whether a communication channel is used as an underground marketplace.

Publications

- K. Krombholz, H. Hobel, M. Huber, and E. Weippl. Social engineering attacks on the knowledge worker. In *Proceedings of the 6th International Conference on Security of Information and Networks*, pages 28–35. ACM, 2013

- K. Krombholz, H. Hobel, M. Huber, and E. Weippl. Advanced social engineering attacks. *Journal of Information Security and Applications*, 22:113–122, 2014

- A. Hudic, K. Krombholz, T. Otterbein, C. Platzer, and E. Weippl. Automated analysis of underground marketplaces. In *IFIP International Conference on Digital Forensics*, pages 31–42. Springer Berlin Heidelberg, 2014

1.4.2 QR Code Security

Our papers on QR Code Security address the challenges of securing information sharing over QR codes in public places where the originator of a posted code is difficult to verify. We decided to focus on this specific scenario where users rely on technology to make an informed decision on whether they are under an attack. QR codes are not human-readable and require technology for decoding and further processing of the decoded content. QR codes are a strong attack vector for phishing attacks as the user cannot distinguish a benign from a malicious QR code by simply looking at it. In [112] we surveyed QR code-based attacks from state of the art literature and identified major research and design challenges. Based on these findings, we presented a security and privacy analysis of the twelve most downloaded QR code reader apps for iOS, Android and Windows Phone in [113]. We found that most apps are not capable of detecting QR code-based phishing attack and furthermore violate user privacy by leaking sensitive information and user tracking and hence leave the burden on the user. Then, we deployed stickers with QR codes in public places in five European cities and measured the performance. Also, they had a link encoded to a short questionnaire. Based on this data, we found that many users perceive such QR codes in public places as phishy, but still scan them. Only half of the participants reported to check an encoded URL before actually visiting it.

Publications

- K. Krombholz, P. Frühwirt, P. Kieseberg, I. Kapsalis, M. Huber, and E. Weippl. Qr code security: A survey of attacks and challenges for usable security. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 79–90. Springer International Publishing, 2014

- K. Krombholz, P. Fruhwirt, T. Rieder, I. Kapsalis, J. Ullrich, and E. Weippl. Qr code security - how secure and usable apps can protect users against malicious qr codes. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 230–237. IEEE, 2015

1.4.3 Privacy in Wearable Computing

Due to the proliferation of mobile phones and tablets, the number of cameras taking pictures in public spaces has amplified. In the coming age of wearable computing, devices with built in cameras and other sensors will become as ubiquitous as smartphones and make instant sharing of pictures and other potentially sensitive information of unintentionally captured bystanders even easier. To date, there is no privacy-mediating tool available off-the-shelf to obtain an informed consent between the photographers and bystanders on how the paper is then used and published online. A handful of approaches has been published at scientific venues but has not evaluated regarding whether they are actually usable and feasible to solve the problem in practice.

Therefore, as a first step, we systematized these approaches and show that already proposed techniques are often not able to prevent bystanders from being recorded without their consent. On the one hand, we found that most approaches require users to actively inhibit filming, which requires them to be aware of being filmed. Especially the discreet recording capabilities of life-logging devices and wearable cameras make it hard for bystanders to notice that they are recorded. Also, about half of the approaches we considered for our systematization were highly visible and therefore potentially impact user behavior. Some approaches propose to have a centralized authorities to enforce individual-chosen privacy policies. These approaches however, potentially violate user privacy as they transmit potentially sensitive information (e.g., biometric features) to third parties.

As usability is a key issue in whether such a system could be deployed on a large-scale to mediate privacy preferences, we derived three conceptual privacy-mediating technologies from related work, i.e., (1) a privacy fabric with a visually encoded privacy preferences, (2) a privacy app that uses a centralized service in the background and (3) a dedicated device in shape of a bracelet. We then conducted semi-structured interviews with 20 participants. Our key findings highlight that there is a high demand for privacy-mediating technology, as cameras in public places are often perceived as disruptive by bystanders. Our results indicate that privacy fabric with visually encoded policies is hard to understand for non-expert users and therefore lacks of acceptance. By far the preferred concept was the privacy bracelet. Our participants reported that haptic tools with a physical button to push give them a sense of control over the currently mediated policy.

Publications

- K. Krombholz, A. Dabrowski, M. Smith, and E. Weippl. Ok glass, leave me alone: Towards a systematization of privacy enhancing technologies for wearable computing. In *Financial Crypto 2015 WEARABLE S&P Workshop*, 2015

Currently Under Submission

- K. Krombholz, A. Dabrowski, M. Smith, and E. Weippl. Exploring design directions for wearable privacy. *Under submission at the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM 2016)*, 2016

1.4.4 User Authentication on Mobile Devices

As modern smartphones store more and more sensitive information about their users it has become increasingly important to protect these devices from unwanted access. Recent studies have shown that many users have weak PINs, passwords or unlock patterns, or do not protect their devices at all. Also, shoulder surfing in public spaces is a major threat which is hard to defeat. Recent smartphone models introduced biometrics such as Android Face Unlock or the TouchID fingerprint sensor for user authentication. However,

they still rely on knowledge-based methods for fallback authentication. Furthermore, biometric methods are non-revocable and can easily be attacked at low cost. The scientific community tried to address these issues by introducing novel authentication methods that have not yet been deployed in off-the-shelf devices. On the one hand, this is due the increased task overhead introduced by these methods. A higher task overhead than the state of the art is infeasible in practice: as a recent study by Harbach et al. has shown that the average smartphone user unlocks their device about 15 times a day. On the other hand, many of these approaches require additional specialized hardware in order to function. In order to address these challenges for future authentication systems, we proposed force-PINs, a knowledge-based authentication scheme that enhances digit PINs with binary pressure values to increase the PIN space. To evaluate this concept, we performed a lab study with 50 users to compare the task overhead to digit-only four- and six-digit PINs, and, a field study over two weeks to show learning effects. We found that force-PINs provide additional security with a minimal impact on usability.

Publications

- K. Krombholz, T. Hupperich, and T. Holz. Use the force: Evaluating force-sensitive authentication for mobile devices. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, 2016

1.4.5 Usability of Cryptographic Applications

The concept of public key cryptography is slowly finding its way into consumer applications. The concept of crypto-currencies heavily relies on these fundamentals, with Bitcoin being the most successful representative to date. Anecdotal evidence has shown that users often struggle with using Bitcoin in a secure and privacy-preserving way even though the concept provides technically solid mechanism in theory. To shed light on the actual situation, we performed a large scale online survey (n=990) and additional qualitative interviews with a subset of 10 participants and covered aspects such as how Bitcoin users manage their keys, how they deal with privacy and anonymity and whether they have lost their keys and/or even money.

Another challenging cryptographic application is Transport Layer Security (TLS), a protocol to secure communications which is to date considered as one of the fundamental building blocks of the Internet. Recent studies have shown that TLS is often poorly configured which leaves services vulnerable to Man-in-the-Middle attacks. When a certificate is missing, the browser issues a warning. These warnings are often hard to understand and therefore suffer from high click-through rates. This phenomena has already been studied and warnings have improved but the problem of vulnearable misconfigurations remains in the Internet ecosystem. Our approach to address this issues was to study TLS usability from an administrators perspective. We found that the TLS deployment process is difficult even for expert users which explains why so many configurations are vulnerable.

Publications

- K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl. The other side of the coin: User experiences with bitcoin security and privacy. In *Financial Cryptography and Data Security 2016*. Springer, 2016

Currently Under Submission

- K. Krombholz, W. Mayer, M. Schmiedecker, and E. Weippl. "i have no idea what i'm doing" - on the usability of deploying https. *Under submission at the Network and Distributed System Security Symposium (NDSS 2017)*, 2017

1.5 Structure of this Thesis

The remainder of this thesis is structured as follows: Chapter 2 presents our taxonomy of social engineering attacks. Chapter 3 demonstrates the results of our machine-learning approach to classify online platforms as underground marketplaces. In Chapter 4, we present our experiments on QR code security and countermeasures to mitigate QR code-based phishing attacks. Chapter 5 contains our research on privacy in the age of wearable computing based on a systematic literature analysis and a user study. In Chapter 6, we present our approach called force-PINs to make user authentication on smartphones more secure but still user-friendly. Furthermore, Chapter 7 presents the findings from a large-scale user study on user experiences with Bitcoin security and privacy. In Chapter 8 we show how poor usability impacts the security of TLS configurations. Finally, Chapter 9 concludes our work and discusses future research challenges.

CHAPTER 2

Social Engineering

Social engineering has emerged as a serious threat in virtual communities and is an effective means to attack information systems. The services used by today's knowledge workers prepare the ground for sophisticated social engineering attacks. The growing trend towards *BYOD* (bring your own device) policies and the use of online communication and collaboration tools in private and business environments aggravate the problem. In globally acting companies, teams are no longer geographically co-located, but staffed just-in-time. The decrease in personal interaction combined with a plethora of tools used for communication (e-mail, IM, Skype, Dropbox, LinkedIn, Lync, etc.) create new attack vectors for social engineering attacks. Recent attacks on companies such as the New York Times and RSA have shown that targeted spear-phishing attacks are an effective, evolutionary step of social engineering attacks. Combined with zero-day-exploits, they become a dangerous weapon that is often used by advanced persistent threats. This chapter presents a taxonomy of well-known social engineering attacks as well as a comprehensive overview of advanced social engineering attacks on the knowledge worker.

This chapter is an extended version of [114] and [115].

The remainder of this chapter is structured as follows: Section 2.1 presents an introduction and Section 3.2 presents the state of the art in social engineering. In Section 2.3 we present our attack taxonomy. In Section 2.4 we discuss two real-life attack scenarios and in Section 2.5 we conclude this chapter.

2.1 Introduction

The Internet has become the largest communication and information exchange medium. In our everyday life, communication has become distributed over a variety of online communication channels. In addition to e-mail and IM communication, Web 2.0 services

such as Twitter, Facebook, and other social networking sites have become a part of our daily routine in private and business communication. Companies expect their employees to be highly mobile and flexible concerning their workspace [22] and there is an increasing trend towards expecting employees and knowledge workers to use their own devices for work, both in the office and elsewhere. This increase in flexibility and, conversely, reduction in face-to-face communication and shared office space means that increasing amounts of data need to be made available to co-workers through online channels. The development of decentralized data access and cloud services has brought about a paradigm shift in file sharing as well as communication, which today is mostly conducted over a third party, be it a social network or any other type of platform. In this world of ubiquitous communication, people freely publish information in online communication and collaboration tools, such as cloud services and social networks, with very little thought of security and privacy. They share highly sensitive documents and information in cloud services with other virtual users around the globe. Most of the time, users consider their interaction partners as trusted, even though the only identification is an e-mail address or a virtual profile. In recent years, security vulnerabilities in online communication and data sharing channels have often been misused to leak sensitive information. Such vulnerabilities can be fixed and the security of the channels can be strengthened. However, even security-enhancing methods are powerless when users are manipulated by social engineers. The term *knowledge worker* was coined by Peter Drucker more than 50 years ago and still describes the basic characteristics of a worker whose main capital is knowledge [57]. The most powerful tool an attacker can use to access this knowledge is *Social Engineering*: manipulating a person into giving information to the social engineer. It is superior to most other forms of hacking in that it can breach even the most secure systems, as the users themselves are the most vulnerable part of the system. Research has shown that social engineering is easy to automate in many cases and can therefore be performed on a large scale. Social engineering has become an emerging threat in virtual communities. Multinational corporations and news agencies have fallen victim to sophisticated targeted attacks on their information systems. Google's internal system was compromised in 2009 [3], the RSA security token system was broken in 2011 [1], Facebook was compromised in 2013 [6], as was the New York Times [150]. Many *PayPal* costumers have received phishing e-mails [171] and many have given the attackers private information such as credit card numbers. These recent attacks on high-value assets are commonly referred to as Advanced Persistent Threats (APTs). APTs often rely on a common initial attack vector: social engineering such as spear-phishing and water-holing. The awareness for software security issues and privacy-enhancing methods has increased as serious incidents have been reported in the media. For example, the awareness for social engineering attacks over e-mail, which is without doubt the most frequently used communication channel on the Internet and is flooded by scammers and social engineers every day, has increased among users. However, the awareness for social engineering in cloud services and social networks is still comparatively low.

The main contributions of this article are the following:

- We discuss social engineering with regards to knowledge workers.
- We provide a taxonomy of social engineering attacks.
- We give an overview of current attack vectors for social engineering attacks.
- We discuss real-world incidents of successful social engineering attacks.

The goal of this chapter is to provide a comprehensive and complete overview of social engineering attacks on the knowledge worker, to monitor the state of the art of research in this field, and to provide a comprehensive taxonomy to categorize social engineering attacks and measure their impact. Our chapter extends the state of the art by including novel, non-traditional attacks such as APTs. Our taxonomy extends and combines already existing work in this field, e.g., by Ivaturi et al. [101] and Foozy et al. [137]. Furthermore, our taxonomy systemizes operators, channels, types and attack vectors as well.

2.2 Background

This section discusses the state of the art of social engineering and computer-supported collaborative work (*CSCW*). Attacks are divided into four different categories: physical, technical, social and socio-technical approaches.

2.2.1 Social Engineering (SE)

Social engineering is the art of getting users to compromise information systems. Instead of technical attacks on systems, social engineers target humans with access to information, manipulating them into divulging confidential information or even into carrying out their malicious attacks through influence and persuasion. Technical protection measures are usually ineffective against this kind of attack. In addition to that, people generally believe that they are good at detecting such attacks. Research, however, indicates that people perform poorly on detecting lies and deception [153, 130]. The infamous attacks of Kevin Mitnick [136] showed how devastating sophisticated social engineering attacks are for the information security of both companies and governmental organizations. When social engineering is discussed in the information and computer security field, it is usually by way of examples and stories (such as Mitnick's). However, at a more fundamental level, important findings have been made in social psychology on the principles of persuasion. Particularly the work of Cialdini [46], an expert in the field of persuasion, is frequently cited in contributions to social engineering research. Although Cialdini's examples focus on persuasion in marketing, the fundamental principles are crucial for anyone seeking to understand how deception works.

2.2.2 Types of Social Engineering Attacks

Social engineering attacks are multifaceted and include physical, social and technical aspects, which are used in different stages of the actual attack. This subsection aims to explain the different approaches attackers use.

Physical approaches

As the name implies, physical approaches are those where the attacker performs some form of physical action in order to gather information on a future victim. This can range from personal information (such as social security number, date of birth) to valid credentials for a computer system. An often-used method is *dumpster diving* [74], i.e., searching through an organization's trash. A dumpster can be a valuable source of information for attackers, who may find personal data about employees, manuals, memos and even print-outs of sensitive information, such as user credentials. If an attacker can gain access to a targeted organization's offices - e.g., in open-plan workspaces - they may find information such as passwords written on Post-it notes. Less sophisticated physical attacks involve *theft* or *extortion* to obtain information.

Social approaches

The most important aspect of successful social engineering attacks are social approaches. Hereby attackers rely on socio-psychological techniques such as Cialdini's principles of persuasion to manipulate their victims. Examples of persuasion methods include the use of (purported) authority. One common social vector that is not explicitly addressed by Cialdini is curiosity, which is, e.g., used in spear-phishing and baiting attacks. In order to increase the chances of success of such attacks, the perpetrators often try to develop a relationship with their future victims. According to [74], the most prevalent type of social attacks is performed by phone.

Reverse social engineering

Instead of contacting a potential victim directly, an attacker can attempt to make them believe that he/she is a trustworthy entity. The goal is to make potential victims approach him, e.g., to ask for help. This indirect approach is known as *"reverse social engineering"* [74, 136] and consists of three major parts: sabotage, advertising and assisting [142]. The first step in this is sabotaging the company's computer system. This can range anywhere from disconnecting someone from the company's network to sophisticated manipulation of the victim's software applications. The attackers then advertise that they can fix the problem. When the victim asks for help, the social engineer will resolve the problem they created earlier while, e.g., asking the victim for their password ("so I can fix the problem") or telling them to install certain software.

Technical approaches

Technical attacks are mainly carried out over the Internet. Granger [74] notes that the Internet is especially interesting for social engineers to harvest passwords, as users often use the same (simple) passwords for different accounts. Most people are also not aware that they are freely providing attackers (or anyone who will search for it) with plenty of personal information. Attackers often use search engines to gather personal information about future victims. There are also tools that can gather and aggregate information from different Web resources. One of the most popular tools of this kind is Maltego[1]. Social networking sites are becoming valuable sources of information as well (see Section 2.4 for more details).

Socio-technical approaches

Successful social engineering attacks often combine several or all of the different approaches discussed above. However, socio-technical approaches have created the most powerful weapons of social engineers. One example is the so-called *baiting* attack: Attackers leave malware-infected storage media in a location where it is likely to be found by future victims. Such *"road apples"* could, e.g., be a USB drive containing a Trojan horse [174]. Attackers additionally exploit the curiosity of people by adding tempting labels to these road apples (storage media), such as "confidential" or "staff lay-off 2014". Another common combination of technical and social approaches is phishing. Phishing is usually done via e-mail or instant messaging and is aimed at a large user group in a rather indiscriminate way, similar to spam. Social engineering, in contrast, is typically directed at individuals or small groups of people. Scammers hope that by sending messages to a vast number of users, they will fool enough people to make their phishing attack profitable. Herley and Florencio [85] argue that classical phishing is not lucrative, which might explain why phishing attacks are moving towards more sophisticated "spear-phishing" attacks. Spear-phishing attacks are highly targeted messages carried out after initial data-mining. Jagatic et al. [102] used social networking sites to mine data on students and to then send them a message that looked like it had been sent by one of their friends. By using such "social data", the authors were able to increase the success rate of phishing from 16 to 72 percent. Hence, spear-phishing is considered a combination of technological approaches and social engineering.

2.2.3 Computer-supported collaboration

Businesses and employees use a wide range of technologies to facilitate, automate and improve daily tasks. We also see collaborative business structures emerging: Computer-supported collaboration tools for file sharing or collaborative workspaces, internal or external communication, blogs, wikis, etc., help connect staff within the company and to

[1] Maltego is an open source intelligence and forensics application. It allows the mining and gathering of information as well as the representation of this information in a meaningful way. `http://www.paterva.com/maltego/`

customers, allow widespread and instant information exchange about the entire business domain, and establish a constant communication channel to the customers and partners of the company.

Considering the wide range of different communication channels created by these computer-supported collaboration tools, social engineering attacks have a huge attack potential. However, in the business context, we differentiate between office communication and external communication. This enables us to make predictions about a victim's ability to detect a social engineering attack.

Office communication

Modern communication tools have changed communication flows among staff members enormously, making the high-speed exchange of information possible. There are sophisticated technologies that protect the security of data transfer. However, the majority of these countermeasures cover technical attacks, while social engineering attacks remain unconsidered. In enterprise environments, face-to-face communication is often replaced by e-mails or instant messages, generating a novel attack surface for social engineers. Obviously, social engineering attacks coming from internal accounts or e-mails with forged internal addresses are more likely to slip through the defenses of a potential victim. For instance, Parsons et al. [146] conducted a role-play scenario experiment in which 117 participants were tested on their ability to distinguish between phishing e-mails and benign e-mails. Their results indicated that people with a higher awareness level are able to identify significantly more phishing e-mails. Valuable personal information gained through social engineering attacks could have direct consequences, such as the exploit of a bank account, or indirect consequences, such as reputation loss [180]; it could also be used to improve the effectiveness of further social engineering attacks. Overall, we face multifarious social engineering attacks - once an attack is successful, the external adversary can use the information to become an insider and perform even more successful social engineering attacks.

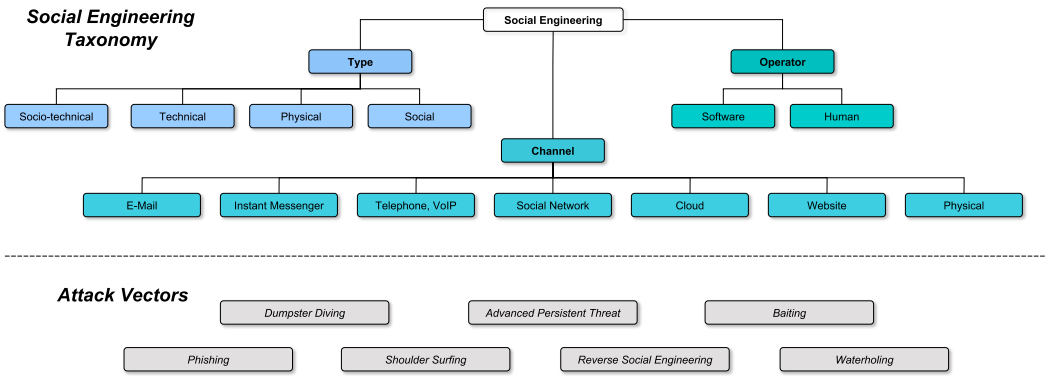


Figure 2.1: Overview of our classification of attack characteristics and attack scenarios.

External communication

As with intra-office communication, there is a trend towards the use of e-mail services, cloud, blogs, etc., for external communication, creating the same challenges as in internal communication. However, as the organizational border becomes increasingly blurred, it is difficult to decide which information may be published or passed on to an external communication partner. For instance, marketing blogs are useful for advertising purposes, but also carry the risk of unwanted information leakage. Another example is the release of information, e.g., about staff members, on LinkedIn, where a potential adversary can find out how many people are employed over a number of years and infer the economic status of the respective company from this data [19]. The strongest potential risk of external communication lies in the broad range of possible communication channels. Furthermore, new trends increase the number of channels, such as Bring Your Own Device (BYOD) [135] and the idea of "technology gets personal", which is used by Thomson [184] to explain the impact of using mobile devices to work with corporate information in insecure environments, such as cafés or public transport systems. He refers to mobile technology as the "window into the enterprises". Of course, security systems are installed on most of these devices; however, these systems offer no protection from social engineering attacks.

2.3 Social Engineering Taxonomy

In this section, we propose a taxonomy for the classification of social engineering attacks. Figure 2.1 illustrates the structure of our taxonomy and the attack scenarios, which we describe in detail in this section.

To classify social engineering attacks, we first introduce three main categories: **Channel**, **Operator**, and **Type**.

Attacks can be performed via the following channels:

- **E-mail** is the most common channel for phishing and reverse social engineering attacks.
- **Instant messaging** applications are gaining popularity among social engineers as tools for phishing and reverse social engineering attacks. They can also be used easily for identity theft to exploit a trustworthy relationship.
- **Telephone, Voice over IP** are common attack channels for social engineers to make their victim deliver sensitive information.
- **Social networks** offer a variety of opportunities for social engineering attacks. Given their potential to create fake identities and their complex information-sharing model, they make it easy for attackers to hide their identity and harvest sensitive information.

- **Cloud** services can be used to gain situational awareness of a collaboration scenario. Attackers may place a file or software in a shared directory to make the victim hand information over.
- **Website**s are most commonly used to perform waterholing attacks. Furthermore, they can be used in combination with e-mails to perform phishing attacks (e.g., sending an e-mail to a potential customer of a bank that contains a link to a malicious website that looks just like the bank's original website).

We also classify the attack by operator. The originator (operator) of a social engineering attack can be:

- **Human**: If the attack is conducted directly by a person. The number of targets is limited due to the lower capacity compared to an attack conducted by software.
- **Software**: Certain types of attacks can be automated with software. Examples include the Social Engineering Toolkit (SET), which can be used to craft spear-phishing e-mails [185]. A number of authors have discussed automated social engineering based on online social networks, such as Boshmaf et al. [37], Huber et al. [95] and Krombholz et al. [119]. The main advantage of automated attacks is that the number of possible targets that can be reached within a short period of time is considerably higher than with purely human attacks.

Furthermore, we categorize social engineering attacks into four types, namely:

- **Physical** as described in Section 2.2.2
- **Technical** as described in Section 2.2.2
- **Social** as described in Section 2.2.2
- **Socio-technical** as described in Section 2.2.2

Concerning social engineering, we determine the following attack scenarios: Attackers perform social engineering attacks over a variety of different channels. They are mostly conducted by humans as well as by software and furthermore categorized as physical, technical, social or socio-technical. The boundaries of the individual types of attack are highly expandable and have, in most cases, not yet been technically exhausted.

- **Phishing** is the attempt to acquire sensitive information or to make somebody act in a desired way by masquerading as a trustworthy entity in an electronic communication medium. They are usually targeted at large groups of people. Phishing attacks can be performed over almost any channel, from physical presence of the attacker to websites, social networks or even cloud services. Attacks targeted

at specific individuals or companies are referred to as *spear-phishing*. Spear-phishing requires the attacker to first gather information on the intended victims, but the success rate is higher than in conventional phishing. If a phishing attack is aimed at high-profile targets in enterprises, the attack is referred to as *whaling*.

- **Dumpster diving** is the practice of sifting through the trash of private individuals or companies to find discarded items that include sensitive information that can be used to compromise a system or a specific user account.
- **Shoulder surfing** refers to using direct observation techniques to get information, such as looking over someone's shoulder at their screen or keyboard.
- **Reverse social engineering** is an attack where usually trust is established between the attacker and the victim. The attackers create a situation in which the victim requires help and then present themselves as someone the victim will consider someone who can both solve their problem and is allowed to receive privileged information. Of course, the attackers try to choose an individual who they believe has information that will help them.
- **Waterholing** describes a targeted attack where the attackers compromise a website that is likely to be of interest to the chosen victim. The attackers then wait at the waterhole for their victim.
- **Advanced Persistent Threat** refers to long-term, mostly Internet-based espionage attacks conducted by an attacker who has the capabilities and intent to comprise a system persistently.
- **Baiting** is an attack during which a malware-infected storage medium is left in a location where it is likely to be found by the targeted victims.

Table 2.1 outlines the relationship between our proposed social engineering taxonomy and current attack scenarios. We classified current social engineering attack scenarios based on our taxonomy. We can, for example, observe that a number of social engineering attacks exclusively rely on a physical attack channel, such as shoulder surfing, dumpster diving and baiting. To protect against this class of attacks, physical security needs to be improved. The table furthermore highlights that the majority of today's social engineering attacks rely on a combination of social and technical methods. Hence, to effectively protect against socio-technical attacks, user awareness for social engineering attacks needs to be improved and their devices protected on a technical level.

2.4 Real-World Attacks

This section provides an overview of state-of-the-art social engineering attacks. These attacks often use personal information from online social networks or other cloud services and can be performed in an automated fashion.

Table 2.1: Classification of social engineering attacks according to our taxonomy.

		Phishing	Shoulder Surfing	Dumpster Diving	Reverse Social Engineering	Waterholing	Advanced Persistent Threat	Baiting
Channel	E-mail	✓			✓		✓	
	Instant Messenger	✓			✓			
	Telephone, VoIP	✓			✓			
	Social Network	✓			✓			
	Cloud	✓						
	Website	✓				✓	✓	
	Physical	✓	✓	✓	✓			✓
Operator	Human	✓	✓	✓	✓			✓
	Software	✓		✓	✓	✓	✓	
Type	Physical		✓	✓				✓
	Technical					✓	✓	
	Social				✓			
	Socio-technical	✓			✓	✓	✓	✓

2.4.1 Online Social Networks (OSNs)

While the more traditional forms of social engineering use information collected through dumpster diving or phone calls, OSNs contain a wealth of personal information that can be misused as an initial source for social engineering attacks. Huber et al. were among the first researchers to argue that OSNs enable automated social engineering (ASE) attacks [93] because information harvested from OSNs is easy to process. The authors showed that information on employees of a given target company can be collected in an automated fashion and potentially misused for automated social engineering. Reverse social engineering describes a particular social engineering technique where an attacker lures the victim into initiating the conversion as described in 2.2.2. Irani et al. [100] argue that OSNs enable reverse social engineering attacks and describe three potential attack vectors. The authors evaluated their proposed attack vectors on three different

OSNs: recommendation-based reverse social engineering on Facebook, demographic-based reverse social engineering on Badoo and visitor-tracking-based reverse social engineering on Friendster. Their results show that reverse social engineering attacks are feasible in practice and can be automated by exploiting the features of current online social networks. While social spam is usually sent via an OSN's primary communication channel, attackers who harvest information can also send traditional e-mail messages to deliver spam because users provide their e-mail addresses on their profiles. If spam is delivered via traditional e-mail instead of OSN platforms, these malicious messages cannot be detected by the OSN's provider. Balduzzi et al. [21] showed that OSNs can be misused for automated user profiling, to validate large sets of e-mail addresses and to collect additional personal information corresponding to these sets.

Social phishing and context-aware spam

Phishing is a widely-spread threat on the Internet and consists of an attacker attempting to lure victims into entering sensitive information like passwords or credit card numbers into a faked website that is controlled by the attacker. It has been shown that social phishing [102], where "social" information specific to the victim is used, can be extremely effective compared to regular phishing. Jagatic et al. [102] found that when phishing e-mails impersonated a target's friend, the success rate increased from 16% to 72%. The social graph is, therefore, not only of value for the social network operator, but also for attackers. This is the case especially if it contains additional information like a valid e-mail address or recent communication between the victim and a friend whom the attacker can impersonate. With automated data extraction from social networks, a vast amount of further usable data becomes available to spammers. Prior conversations within the social network, such as private messages, comments or wall posts, could be used to determine the language normally used for message exchange between the victim and his friends, as a phishing target might find it very suspicious to receive a message in English from a friend with whom they normally communicate in French. Context-aware spam misuses personal information extracted from OSNs to increase the appearance of authenticity of traditional spam messages. Brown et al. [40] identified three context-aware spam attacks: relationship-based attacks, unshared-attribute attacks, and shared-attribute attacks. Relationship-based attacks solely exploit relationship information, making this the spam equivalent of social phishing. The two other attacks exploit additional information from social networks, information that is either shared or not shared between the spam target and the spoofed friend. An example of an unshared attack are birthday cards that seem to originate from the target's friend. Shared attributes, e.g., photos in which both the spam target and her spoofed friend are tagged, can be exploited for context-aware spam. Huber et al. [94, 96] found that the missing support for communication security can be exploited to automatically extract personal information from online social networks. Moreover, the authors showed that the extracted information could be misused to target a large number of users with context-aware spam.

Fake profiles

At the time of writing, the only requirement for the creation of a social networking account is a valid e-mail address, which makes it rather easy for attackers to create fake accounts. A study by Sophos published in 2007 with randomly chosen Facebook users showed that approximately 41% of social networking users accepted friendship requests from a fake profile [173]. Ryan and Mauch [8] further showed that fake profiles can be misused to infiltrate social networks: they set up a profile for a fictional American cyber threat analyst, called "Robin Sage", and were able to gain access to sensitive information in the military and information security community. Bilge et al. [29] outlined two sophisticated fake profile attacks that could be used to infiltrate the trusted circles of social networking users: profile cloning attacks, where attackers clone existing user profiles and attempt to "reinvite" their friends, and cross-profile cloning attacks, where attackers create a cloned profile on an online social network where the target user does not yet have a profile and then contact the targets' friends. If a user, for example, has a Facebook account but no LinkedIn account, an attacker could clone the Facebook profile to create a LinkedIn profile and then contact the target's Facebook friends who are also on LinkedIn. Bilge et al. showed that their attacks can be fully automated and are feasible in practice. If an attacker is able to create fake accounts on a large scale, Sybil attacks on OSNs are possible. OSN providers therefore use various protection mechanisms to limit the creation of large amounts of fake accounts [175]. Boshmaf et al. [37] however found that OSNs can be infiltrated on a large scale. They evaluated how vulnerable OSNs are to a large-scale infiltration by socialbots - computer programs that control OSN accounts and mimic real users. The authors created a *Socialbot* Network (SbN): a group of adaptive socialbots that are orchestrated in a command-and-control fashion on Facebook. The authors used 102 fake profiles to send friendship requests to 5,053 randomly selected Facebook users. 19.3% of these users accepted the friendship requests. Next, the SbN tried to infiltrate the circle of friends of the users who had accepted their fake friendship requests. Within 8 weeks, the SbN was able to further infiltrate the network and gain access to personal information. A recent survey by Alvisi et al. [15] provides an overview of Sybil defenses for online social networks and proposes community detection algorithms.

2.4.2 Cloud services

Cloud services provide a new channel through which social engineers can conduct attacks on the knowledge worker. Knowledge workers frequently collaborate with others who do not work at the same location. Sharing information on a cloud service has therefore become popular. In this scenario, an attacker exploits this situation and uses the cloud as a channel for the social engineering attack. Recent publications described a variety of possible attacks in the cloud, e.g., an attacker placing a malicious file into another user's cloud as described by Gruschka et al. [75] and then using social engineering to make them execute the malicious file. A malicious piece of software can also be used to extract personal information from the victim's account, which is then used to perform more

targeted attacks. Mulazzani et al. [140] provide countermeasures to reduce the risk by preventing the attacker from placing malicious files on Dropbox, one of the currently most commonly used cloud services. The level of trust between users of a shared directory or file is not always as high as desired. Social engineers can exploit this fact by using a fake identity or a compromised user account to invite the victim to share specific information with the attacker in the cloud. According to Roberts et al. [158], one of the biggest weaknesses of cloud services is that the users - companies and individual users - lose control over their data when they store and access it remotely. On traditional servers that are owned by a company itself, it can restrict access and define customized access policies. In cloud services, the responsibility for that is shifted to a third party. Therefore, if a cloud service is to be used for the exchange of sensitive information, a certain level of trust must be established not only between collaborating users, but also between the cloud hosting company and the user. The most commonly observed attacks on cloud services are spear-phishing and APTs.

2.4.3 Mobile applications

The increased use of mobile applications in both business and private contexts makes them an increasingly popular channel for social engineering attacks. In business communication, mobile messaging and e-mail applications are of high interest to social engineers. *BYOD* policies established by companies often include the use of mobile phones and tablets. More and more employees use their smartphones to check their company e-mails or to read documents that are stored in the cloud. However, many smartphone users use highly vulnerable smartphone applications that can be misused to conduct social engineering attacks. Schrittwieser et al. [162] presented two different attack scenarios that can serve as a starting point for such an attack. In their work [162], they demonstrated how sender ID spoofing can be done on popular mobile messaging applications such as *WhatsApp* [10]. A social engineer can use this to send a message to a victim while pretending to be one of his friends. The authors also highlighted how vulnerabilities can be exploited to hijack user accounts, which can then be used to perform social engineering. Considering that many smartphone applications are highly vulnerable and can leak sensitive information, we can conclude that such mobile devices offer a variety of attack vectors for social engineering and other attacks on user privacy. Moreover, some smartphone applications request permissions to access sensitive data on the user's device. If an attacker were to create such an application, they would obtain the information and could use it as a starting point for a social engineering attack. Chin et al. [44] discussed how inter-application information exchange can be sniffed on smartphones and then be misused to violate application policies and permissions. In some cases, such as described by Potharaju et al. [151], the attacker simply plagiarizes a popular smartphone application and deploys it in order to perform an attack.

2.5 Conclusion

In this chapter, we described common attack scenarios for modern social engineering attacks on knowledge workers. *BYOD*-policies and distributed collaboration as well as communication over third-party channels offers a variety of new attack vectors for advanced social engineering attacks. We believe that a detailed understanding of the attack vectors is required to develop efficient countermeasures and protect knowledge workers from social engineering attacks. To facilitate this, we introduced a comprehensive taxonomy of attacks, classifying them by attack channel, operator, different types of social engineering and specific attack scenarios. We discussed real-world examples and advanced attack vectors used in popular communication channels and the specific issues of computer-supported collaboration of knowledge workers in the business environment such as cloud services, social networks and mobile devices as part of *BYOD* policies. We not only discussed complex advanced attack scenarios, but also provided a comprehensive classification that can serve as a basis for the development of countermeasures and further interdisciplinary research in the field.

CHAPTER 3

Underground Marketplaces

Cyber crime, such as theft of credentials or credit card fraud has emerged as a new type of crime in recent years. Cyber criminals usually attack Internet services to steal sensitive data and operate in crowded online underground marketplaces. Crime investigators and digital forensics are trying to detect and analyze these marketplaces. However, due to the lack of efficient and reliable methods to detect underground marketplaces, investigators have to analyze those channels manually. This is a complex and time-consuming task that is associated with high financial costs. In this chapter, we demonstrate how machine-learning algorithms can be efficiently used to automatically determine whether a communication channel is used as an underground marketplace. Our approach includes specific design features related to the context domain of cyber crime and can be used to reliably detect and observe marketplaces of the underground economy. The manual effort is significantly reduced, leading to lower financial costs, less time required and higher efficiency. We implemented a prototype that classified 51,3 million message samples correctly which implicates that machine learning can be efficiently used for a forensic analysis of underground marketplaces.

This chapter is an extended version of [97].

The remainder of this chapter is structured as follows: In Sections 3.1 and 3.2 we provide background information. We present related literature in Section 3.3. In Section 3.4 we present our framework which is evaluated in Section 3.5. Section 3.6 concludes our work.

3.1 Introduction

In recent years, cyber criminals have established a thriving underground economy over the Internet. They routinely use *underground marketplaces* to communicate and to trade stolen or illegal goods and services. Typically, publicly accessible chatrooms or Web

forums are used as marketplaces where criminals openly market their goods and initiate trade agreements.

Furthermore, recent research [90] has shown that underground marketplaces also have a significant impact on security because they are heavily affected with impersonation attacks to steal credentials, credit card numbers or other sensitive data forgery. Forensic investigators put high manual effort in gaining a deeper understanding of the interdependencies between individual marketplaces and the underground market's participants. Detecting these trading hubs is a tedious and time-consuming task. Clearly, automatically locating underground marketplaces would improve the capability of forensic analysts to acquire real-world data on the underground economy. Unfortunately, the large number of online marketplaces and their *ad-hoc* nature and volatility prevent *naïve* detection approaches, such as simple Web crawling systems from being effectively used. Furthermore, criminals often "hijack" benign websites (e.g., websites that contain classified ads or abandoned forums) instead of using dedicated underground websites.

In this work, we demonstrate how machine learning can be efficiently used a method in digital forensics to automatically detect underground marketplaces. We implemented a prototype and performed an experimental evaluation based on real-world communication channels. We evaluated our methods on data that had been extracted during a period of eleven months from real-world IRC (Internet Relay Chat) rooms and Web forums. Our results show that our system is able to successfully and automatically find and monitor communication channels that are used by cyber criminals and therefore can be used within a digital forensic analysis. We also compare several classification methods and conduct runtime measurements to show the efficiency of our system. The data that we collected from underground communication channels can lead to new findings on cyber crime or even identify ongoing criminal activity. In summary, the main contributions of this work are:

- A proof-of-concept implementation to demonstrate that machine learning can be efficiently used during a forensic investigation to detect and analyze underground marketplaces.
- An empirical evaluation of real-world data that we extracted from IRC channels and Web forums during a one year period.
- We measured and evaluated the performance impact of our technique and present our results.

3.2 Background

3.2.1 Underground Marketplaces

While in theory any type of communication channel could be used as an underground marketplace, only two types are prevalent in reality: IRC chatrooms and Web forums. Clearly, neither one is solely used for cyber crime. In fact, both are popular as they have

a multitude of legitimate use cases. The ability to automatically determine whether a specific instance of a communication channel is related to cyber crime would clearly be beneficial to forensic investigators.

3.2.2 Collecting Data

Acquiring reliable data from underground marketplaces [90, 197] has become a heavily investigated topic for researchers. While the importance of these marketplaces seems obvious, most academic publications [154] are focusing on content evaluation of messages in these marketplaces instead of the methodology for collecting the data from them.

Currently, finding underground marketplaces is a complex and time-consuming manual task. To automate this process, we proposed a novel classification method to discover and subsequently monitor underground marketplaces, even if they are hidden among seemingly benign information channels. Gathering a substantial amount of data from multiple independent underground sources can provide security researchers with valuable data and insights that support the fight against cyber crime. Studying the underground economy has an enormous impact on data security. A major reason for this is that in most cases credit card numbers are traded, which leads to credit card frauds where money is stolen from the victim's bank account. Our classification system can be used alongside existing systems for monitoring the underground economy, such as [64]. As any real-world implementation of such a system will have limited computing and networking resources, it is clearly beneficial to be able to automatically focus on monitoring interesting channels. Additionally, we refined our system in a way to minimize the classification of benign chatrooms as "suspicious" in order to prevent monitoring of non-underground information channels.

3.2.3 Vector Space Model

For our classification system, we initially map the *terms* from each *document* to a numeric vector representation. In our problem domain we define a document as either an IRC chat room or a Web forum (thread) and the terms are the content of the associated messages or posts. We used the *bag of words* (BOW) model [80] to represent each document in the vector space. This model is agnostic to the exact ordering of terms within a document and interprets the terms as a set for each document. The implied *vector space model* allows different weightings of the frequency of individual terms. In the following, we introduce the weighting process of the terms in a document and show why a dimensionality reduction of the vector space is necessary.

Term Frequency and Weighting

For the weighting of terms in a document, we used the *tf-idf* (term frequency - inverse document frequency) [160] approach. The term frequency $tf_{t,d}$ represents the frequency of term t in the document d, whereas the inverse document frequency idf_t indicates the

importance of term t to the document corpus. Together, the term frequency and the inverse document frequency method complement the *tf-idf* weighting scheme.

The *tf-idf* weighting scheme reduces the impact of common words, i.e., those with a high frequency within the document. For the comparison of documents with different lengths we used the well-known *cosine normalization* [170] add-on smoothing to allow a post expansion of the feature space.

Similarity and Distance

A common approach for computing the similarity between two documents represented in the vector space is defined by the *cosine similarity*. The definition is shown in Equation 3.1.

$$sim(d_1, d_2) = \cos\theta = \frac{\vec{V}(d_1) * \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} \tag{3.1}$$

The cosine similarity compensates the document length via the well-known *cosine normalization* and measures the similarity of the relative distribution of the terms by finding the cosine between the two document vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$. The cosine of the angle θ between the two document vectors ranges from one to zero, where one means that the two document vectors are identical and zero indicates that they are independent. The cosine similarity cannot become negative because of the non-negative term frequency, where normally minus one would mean the exact opposite of the other vector. Another conventional measure for the similarity of two vectors is the *Euclidean distance*. This approach is more appropriate if the length of the documents is considered. For example, the Euclidean distance measure is used to compute the nearest neighbors or, in our case, to determine the centroid of the cluster during the document selection process.

Feature Selection

Feature selection describes the process of selecting a subset of terms from the training set that is used for the vector space model. This is an important process because a vector space with small cardinality significantly reduces computation time, whereas a reduction of "noisy" features will increase the accuracy of the classification results.

For a large document corpus, the vector space model relies on a high dimensional vector space, where each document is represented by a *sparse vector*.

In our prototype system, we eliminate noisy features in the filtering stage of the preprocessing phase. In particular, we remove features with an occurrence of less than three times in the training set of the document corpus, as proposed by Joachims et al. [103] to refer *Luhn's* model [125]. For the feature selection, we also calculate the ranked *Information Gain* (IG) of each term t with regard to the class c, as shown in Equation 3.2, where H denotes the entropy. As a result, we can reduce the feature space to one fifth of the size.

$$IG(c,t) = H(c) - H(c|t). \tag{3.2}$$

Selecting terms exclusively from the target class works well for high precision classification results, while selecting terms according to the information gain produces more accurate results. An example of the IG-based feature selection where the top 15 word 4-grams from our IRC data collection are outlined. The sample in line is a request to a service bot with the nickname "chk", which validates credit card information passed as arguments (credit card number, expiration date and card verification code (CVV) marked by the tagger).

3.2.4 Document Selection

The acquisition process as shown in Figure 3.1, provides afterwards a significant reduction of documents in the training set and therefore reduces the necessary human effort.

The document selection is based on *hierarchical agglomerative clustering* (HAC), a frequently used deterministic bottom-up clustering algorithm that does not require the pre-specified number of clusters as input. HAC merges documents with the highest similarity into a cluster. The similarity of a merged cluster is called the *combination similarity*. Our HAC prototype implementation supports *single-link* and *complete-link* clustering. Single-link clustering defines the combination similarity by the most similar members, the merge criterion is therefore local. Complete-link clustering, on the other hand, defines the similarity of two merged clusters by the similarity of the most dissimilar members and merges to a non-local criterion. The algorithm merges documents into clusters until a predefined cutoff similarity value is reached.

3.2.5 IRC and Web forum classification

With regard to our problem domain, the learning algorithm of the classifier approximates the optimal function $f : D \to C$ that maps all document vectors D to the specified class $c \in C$ based on the training set.

Our system implementation currently supports the SVM-Light classifier [103] and a set of classifiers provided by the Weka [191] machine learning toolkit. In our evaluation, we use SVM-Light with a linear kernel function and default parameters, which performed best in our initial experiments compared to other machine learning methods from Weka like *Naïve Bayes (NB)*, *IBk* (a k-nearest neighbor classifier), *SMO* (which implements the sequential minimal optimization algorithm), or the J48 algorithm, which is based on a pruned C4.5 decision tree.

3.2.6 Internet Relay Chat

A large number of publicly accessible Internet Relay Chat*IRC networks* can be found on the Internet (e.g., QuakeNet, IRCnet, Undernet, EFnet, Rizon, Ustream, IRC-Hispano,

etc.). In most cases, they don't requier any access privileges or authentication mechanisms from the user's side, which, unfortunately, does not guarantee reliability. Cyber criminals exploit the benefits of IRC for free advertising of their goods and services. While some IRC networks appear to be specifically designated for cyber crime, benign networks are often abused by criminals as well. They simply create channels with names that are known by insiders to be crime-related. For example, channels with names that start with "*#cc*" (short for "credit card") are often related to criminals that focus on credit card fraud.

In addition to IRC channels, cyber criminals often operate underground marketplaces on websites that contain forums and message boards. These forums organize their content in *threads*, i.e., lists of messages that belong to the same topic. In *Web forum* terminology, a message is usually called *post*. In contrast to IRC, the content of these forums remains persistently published and they allow users to communicate in a more organized way, e.g., by replying to specific posts or to groups of users. Forums generally have stricter admission procedures than IRC (e.g., users have to sign-up to receive login credentials) and also offer "convenience" services to their members, for example, escrow services or private messaging functionality.

3.2.7 Towards an Automated Underground Economy Detection System with Machine Learning

Currently, investigations of underground economy marketplaces involve complex manual data collection and analysis procedures. To identify underground marketplaces, we collected messages from real-world IRC channels as well as Web forums and proposed a text-based classification method. Text classification [164] is the process of labeling texts with a predefined set of attributes to determine class membership. During the learning or training phase of the system, a classifier is derived from the training data that decides class membership when using the system.

Our work emphasizes the benefits of machine learning mechanisms in digital forensics., We demonstrate this by applying the mechanisms that automatically detect underground marketplaces in arbitrary information channels. Additionally, machine learning reduces human effort, flexibly increases the scope of data acquisition, significantly decreases the amount of data, mitigates the human error rate, and prevents malicious data distribution. The variety combination of classification and analysis methodologies ensure a more precise and accurate results, and also iterative analysis.

We apply well-known text and data mining techniques, namely information retrieval[129] and automated text categorization[164]. We successfully combine these techniques with the vector space model-based classification system in order to analyze the information retrieved from chat rooms and Web forums.

Our system implementation is designed as a flexible framework, and therefore each individual component can easily be adopted. The flexibility of our design allows us to use various system configurations incorporated with different components and techniques for

data preprocessing and building a flexible vector space mode, or even applying different classification methodologies. So far, our implementation only supports the two most commonly used information channels among cyber criminals (IRC and Web forums) to demonstrate the feasibility and efficiency of our approach. Therefore it is feasible to extend our implementation to support other communication channels. Furthermore, our framework provides a method to simultaneously create multiple classification processes. This is a particular benefit for multi-label classification to efficiently assign multiple subcategories to the information channel content.

3.3 Related Work

Researching the underground economy is not a new topic, and several related studies have been published in the last years.

Franklin et al. [67] performed a systematic study of IRC channels exploited as underground marketplaces. They evaluated the content by exploiting machine learning techniques and showed that underground marketplaces had considerably implications for Internet security. Furthermore, they analyzed and presented possible approaches for disrupting underground marketplaces. Symantec presented a significant amount of data from IRC and Web forums captured during a period of one year in their study [179], but the authors do not provide any detailed information about the methodology they used to collect and analyze the data. The study by Thomas and Martin [183] mainly focused on the structure and players of the underground economy by examining IRC-based marketplaces. They exposed the information about the infrastructure that the criminals had established as well as the associated activities, alliances, and advertisement methods of underground markets.

Zhuge et al. [197] presented an overview of the underground market and malicious activities on Chinese websites based on a black market bulletin board and an online business platform. The authors focused their study on malicious webpages. Holz et al. [90] presented a different approach, pointing out the impact of the underground economy by analyzing data on “dropzones” that trade stolen digital credentials. They evaluated the method, which enables automated analysis of impersonation attacks.

In contrast to the previous studies, Herley and Florencio [86] argue that marketplaces such as IRC channels and Web forums do not have a significant impact and described them as a standard example of a market for lemons where the goods are hard to monetize and the only people who benefit are the rippers.

Fallmann et al.[64] presented a novel system for automatically monitoring IRC channels and Web forums. Furthermore, they extracted information and performed an experimental evaluation of the monitored environments.

3.4 Framework Model

3.4.1 System Model

We depict our model through two essential classification lifecycle processes, namely the *training process* and the *classification process*.

Training Process

To construct a reliable and efficient classifier, we carefully chose a set of training data using k-fold cross-validation. The raw training data contains noise and content that is not relevant for classification. Therefore, text preprocessing is the initial stage of the training phase. We designed the text preprocessing module according to the pipes-and-filters architecture pattern [134]. The overall goal of this task is to extract the plain text content of the information channel. For this purpose, the representational specifics of the information channel are considered. HTML elements and specific character encodings are eliminated. Within the preprocessing step the textual content has to be prepared for the mapping into the vector space domain. This vector space transformation is performed using tokenization ([133]) to separate chunks of text with a specific semantic value. In our system, we used a word-based model, as it has the best performance according to state of the art research [164], [20]. The next step within the vector space transformation towards a vector representation is the tagging of semantically meaningful units that carry domain-relevant information. In our case, the selected context domain is underground economy. For this purpose, we attached different labels to Uniform Resource Identifiers (URIs), domain names, IP addresses, e-mail addresses and types like numbers and dates to be able to identify the content. The tagging process benefits the feature space reduction by removing frequently changing date values or substituting them with a tag label. The next step is the selection of appropriate documents according to their relevance in the given context in order to reduce the amount of documents in the training set. Our document selection process currently supports two methods for choosing the representative for each cluster: The first selects the document that represents the centroid based on the Euclidean distance, while the second is based on a definable score function. The training set is determined based on the selected documents and appropriate features and their weights, which are specific for representatives of the associated classes, are selected to retrieve a subset of terms from the training set to be used for the vector space model. The reduction of noisy features enhances the accuracy of the classification results. The vector representation is adapted according to the selected features by modeling training instances from the training set. A classifier is constructed and the classifier model is processed. Figure 3.1 illustrates the whole training process.

Classification Process

Figure 3.2 shows a schematic overview of the classification process. The classifier obtained from the training phase is applied in the productive environment of the system. First, the productive input data is prepared for classification. The initial stage is text preprocessing

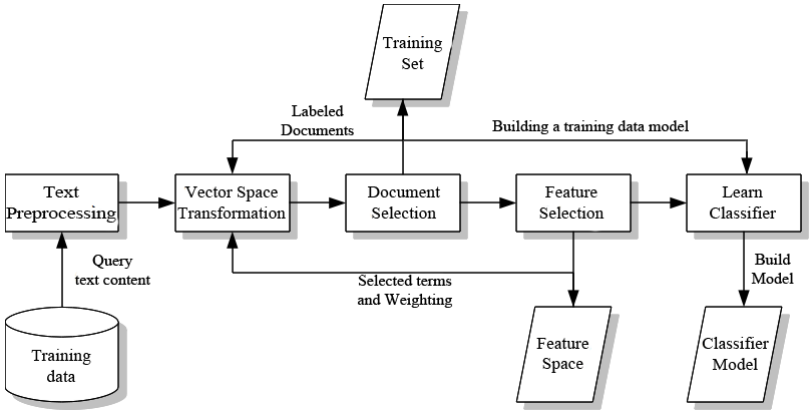


Figure 3.1: Schematic overview of the training process.

to extract the information of interest just like in the training phase. The data is also transformed to a vector space model. Finally, the corresponding features are weighted according to the feature space model and classified using the classifier model built in the training phase. The results are a prediction of class membership of the according input data.

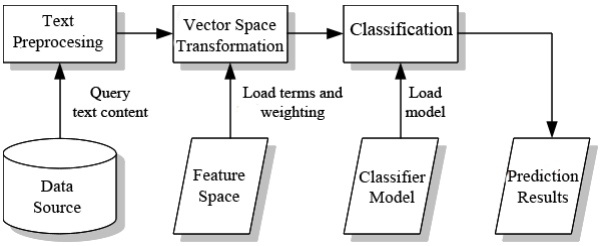


Figure 3.2: Schematic overview of the classification process.

3.5 Evaluation

We used our classification-based approach to detect underground marketplaces in suspicious information channels and extract relevant information. Furthermore, we also compared different vector space models and evaluated the effectiveness of the document selection. Our data corpus was collected over a period of eleven months via an observation framework [64]. During this period, we managed to capture 51.3 million IRC messages transmitted over 2,693 channels on 246 networks. For the Web forum evaluation, we crawled the content of more than 203,000 threads in ten forums. First, we outlined the data collection for our results and explained the differences between performance indicators. Then we evaluated the performance of the classification system in detecting underground marketplaces in the presented data collection for IRC channels and Web forums.

Performance Evaluation of IRC Channels

For the performance evaluation on IRC, we manually labeled all 2,693 IRC channels regarding their relationship to the underground economy and performed the k-fold cross-validation on all of them. Figures 3.3, 3.4, 3.5, and 3.6 show the cross-validation results for underground marketplace detection in IRC

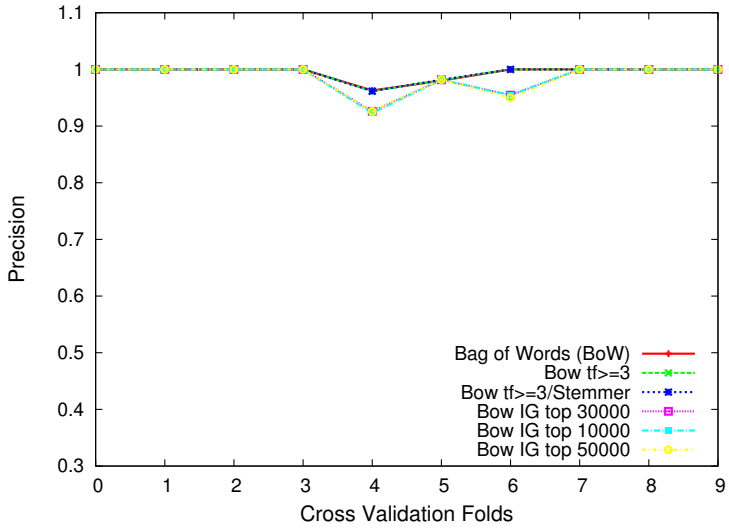


Figure 3.3: Classification precision.

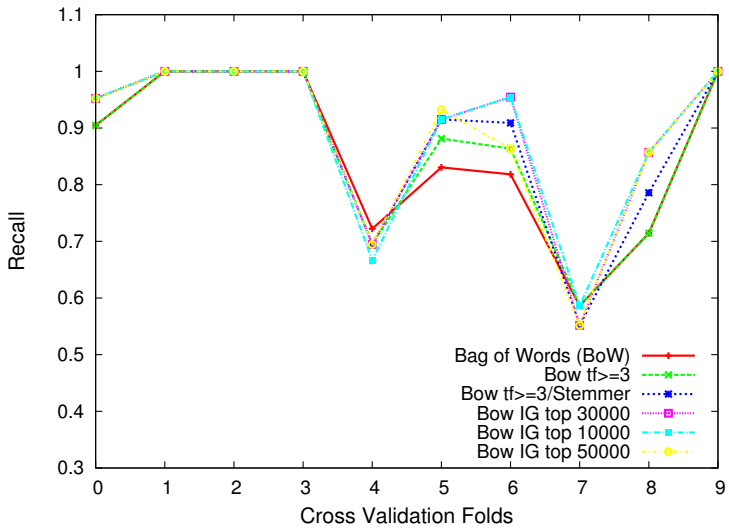


Figure 3.4: Classification recall.

Figure 3.3 shows that our SVM classifier maintains a constantly high precision rate, which means that the predicted results do not contain any false positives. The loss on

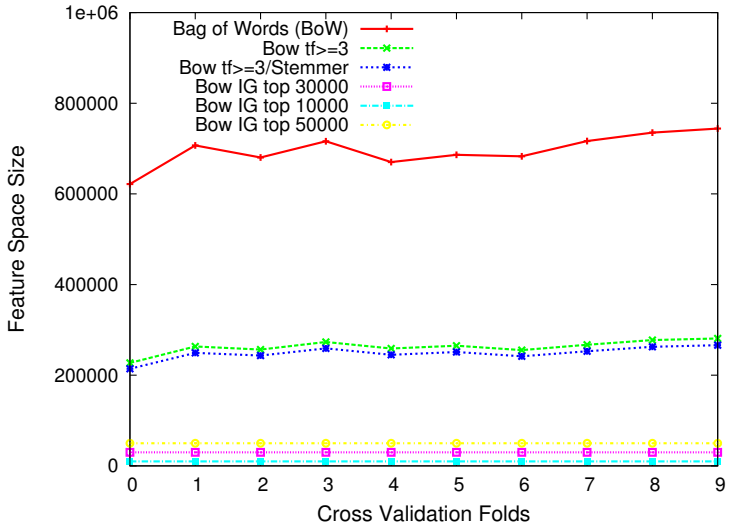


Figure 3.5: Feature space size.

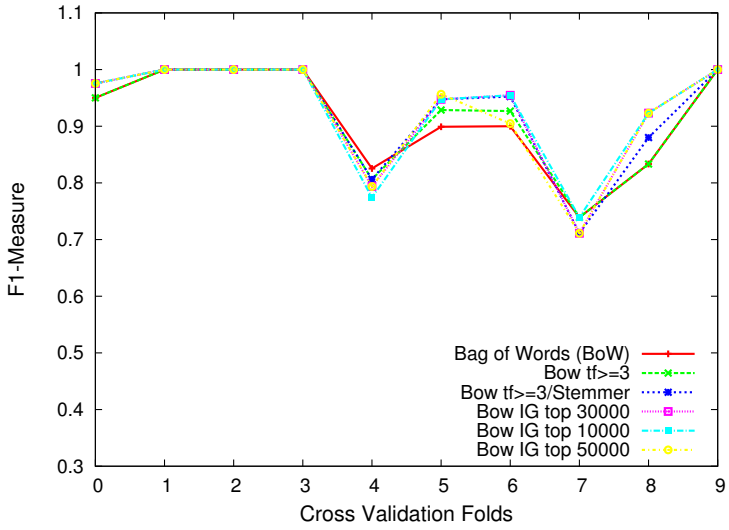


Figure 3.6: Classification F_1-measure.

the recall rate in Figure 3.4 can be mostly attributed to channels in which underground economy-related content accounts for only a fraction of the exchanged messages and will therefore mistakenly be classified as a false negative. In general, removing terms with a $tf < 3$ combined with the English stop word list and the Porter stemmer produced an average precision of 99.43% and increased the recall from the initial 85.76% to an average of 88.09%. The feature selection based on the top 10,000 ranked by the IG reduced the vector space to 4% of the noise filtered space and had the best score with an average precision of 98.59% and a recall of 89.32%. This leads to an average F_1-measure

of 93.14% and an average accuracy of 97.84% and shows that our classification system performs very well on the (noisy) content of IRC channels.

Additionally, we evaluated the performance of the document selection in relation to different similarity values. To this end, the IRC channels are merged to clusters determined by the specified combination similarity cutoff value. The document selection evaluation also analyses the selection methods for the cluster representative and compares the centroid-based method against the score function approach, which is defined by the ratio of unique textual content to the number of messages in the channel.

We performed k-fold cross-validation based on the training sets generated by the document selection. The average performance results of document selection in IRC channels are shown in Figures 3.7, 3.8, 3.9, 3.10.

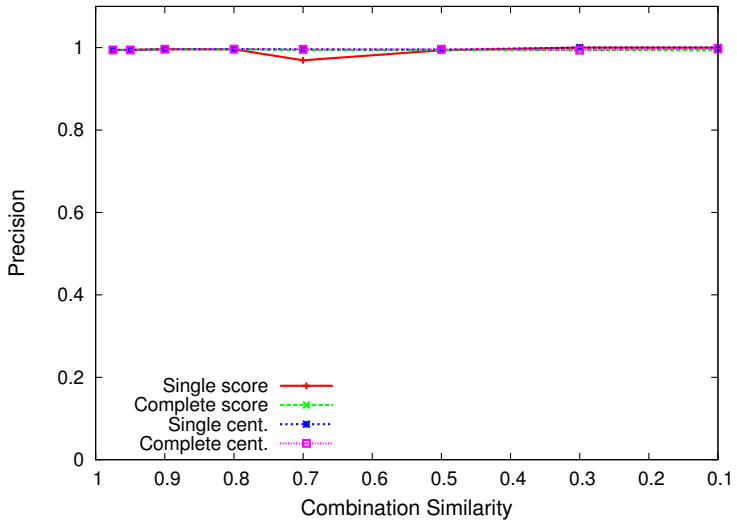


Figure 3.7: Classification precision

While single-link clustering reduces the number of clusters for the given similarity values more rapidly, it also produces a significant loss of accuracy. In contrast, complete-link clustering could reduce the number of needed training samples to less than 40% with a minimal loss of recall. As shown in Figure 3.10, the selection methods for the cluster representative, which will be added to the training set, performed equally well for the upper interval of the combination similarity. At the end, the deviation of the two methods is only visible for a very low combination similarity where the score function based on the content information performed slightly better.

Performance Evaluation of Web Forums

To evaluate the performance of the classification system on Web forums, we manually labeled 300 randomly selected threads from the Web forum *www.clicks.ws* on whether their posts were related to the underground economy or not. In addition, we extended

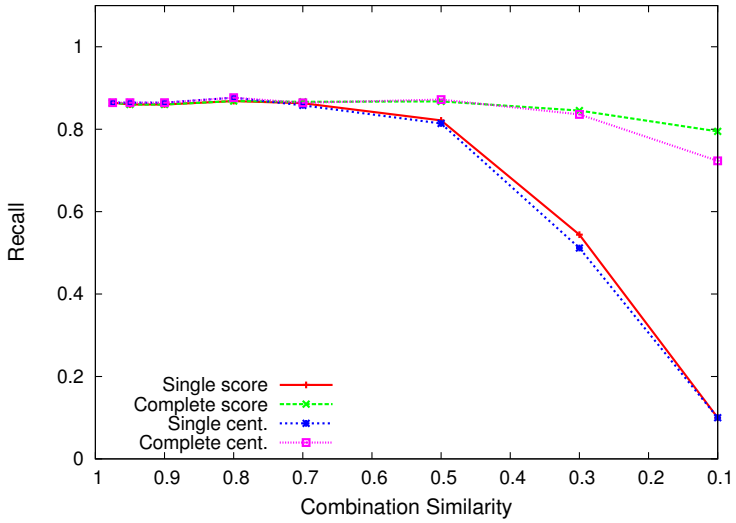


Figure 3.8: Classification recall.

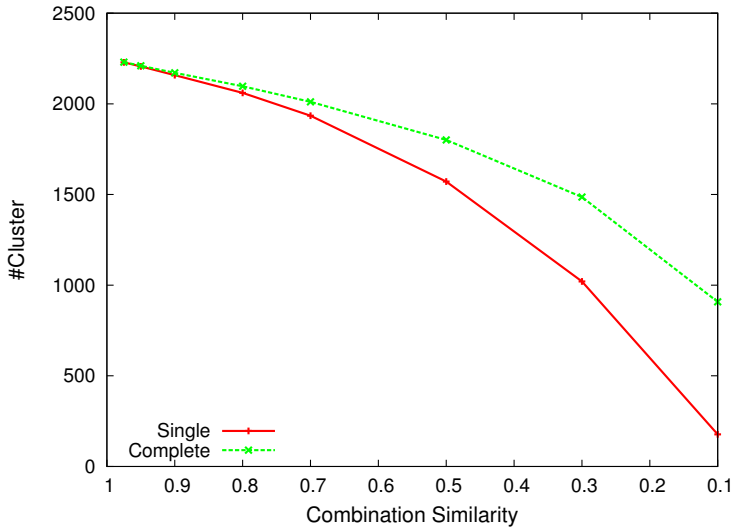


Figure 3.9: Number of clusters.

the training set by another 100 randomly selected threads from each of the other nine Web forums. Table 3.1 shows the average performance of the classification system on the k-fold cross-validation of the Web forum test set.

Our classification system performs very effectively on the Web forum threads, but unfortunately not quite as effectively as with IRC channels. The content of IRC channels shows a more structured discussion and the threads are less noisy, which makes it easier to extract the information. The loss of accuracy is mostly caused by the dissimilarity

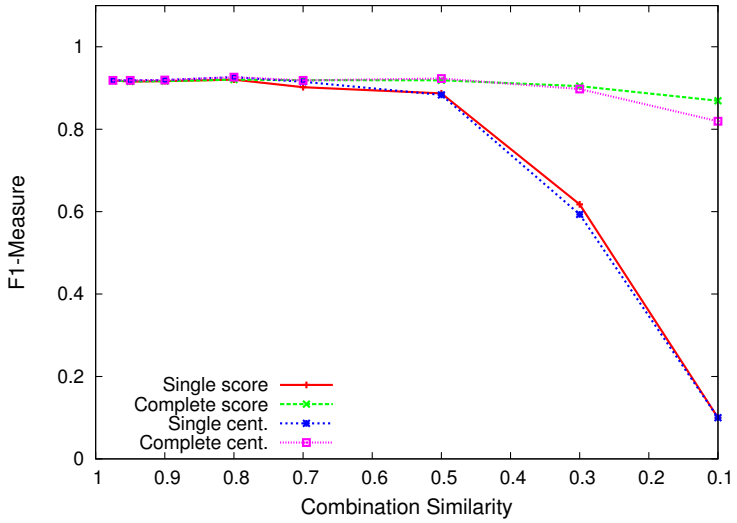


Figure 3.10: Classification F_1-measure.

	Space size	Precision	Recall	Accuracy	F_1-measure
BOW	⌊34,890⌋	96.79%	83.55%	94.02%	89.4%
BOW, $tf < 3$	⌊14,391⌋	96.95%	83.75%	94.2%	89.72%
BOW, $tf < 3$, Stemmed	⌊11,720⌋	97.22%	84.58%	94.38%	90.32%
BOW, IG Top 5,000	5,000	95.87%	83.04%	94.01%	88.9%
BOW, IG Top 3,000	3,000	94.66%	81.6%	93.38%	87.37%
BOW, IG Top 1,000	1,000	94.81%	82.31%	93.47%	87.93%

Table 3.1: Average results of classification performance on Web forums.

of the selected samples, especially due to the German Web forum *www.carders.cc*. As highlighted in Table 3.1, the approach with $tf < 3$ and English stop word filtering combined with the Porter stemmer performed best with an average F_1-measure of 90.32%. The IG-based feature selection could not show its advantages but is clearly not necessary in this case with regard to the dimension of the vector space. In conclusion, the vector space models show similar behavior as in the IRC channel evaluation, demonstrating that our system is capable of effectively detecting Web forums that are used by cyber criminals.

3.6 Conclusion

In this chapter, we demonstrated how text-classifier can be successfully used as a tool to detect and analyze underground marketplaces. Automatically identifying (and monitoring) such marketplaces is important, as it allows forensic analysts to investigate online crime and to acquire data from related sources such as chatrooms that are used by cyber criminals.

Our machine learning-based classification system includes specific design features related to the domain of cyber crime to automatically and reliably detect underground marketplaces in IRC channels and Web forums. This significantly reduces the amount of human interaction necessary for finding such information sources. The prototype system was capable of detecting underground marketplaces with an average accuracy of 97% in a collection of 51.3 million IRC messages, spanning a time period of approximately one year. Furthermore, we were able to classify a subset of threads from ten different Web forums, ranging from underground economy discussion forums to hijacked benign Web forums, with an average accuracy of 94 %.

This demonstrates that our system can effectively be used in a real-world setting to automatically and reliably detect underground marketplaces in suspicious information channels.

CHAPTER 4

Usable Security of QR Codes

QR codes have emerged as a popular medium to make content instantly accessible. With their high information density and robust error correction, they have found their way to the mobile ecosystem. However, QR codes have also proven to be an efficient attack vector, e.g. To perform phishing attacks. Attackers distribute malicious codes under false pretenses in busy places or paste malicious QR codes over already existing ones on billboards. Ultimately, people depend on reader software to ascertain if a given QR code is benign or malicious. In this chapter, we present a comprehensive analysis of QR code security. We determine why users are still susceptible to QR code based attacks and why currently deployed smartphone apps are unable to mitigate these attacks. Based on our findings, we present a set of design recommendations to build usable and secure mobile applications. To evaluate our guidelines, we implemented a prototype and found that secure and usable apps can effectively protect users from malicious QR codes.

This chapter is an extended version of [112] and [113].

The remainder of this chapter is structured as follows: Section 4.1 provides an introduction and Section 4.2 related work. In Section 4.3 we present our evaluation of apps and in Section 4.4 we present an intercultural user study. We discuss our results in Section 4.5 and conclude them in Section 4.6.

4.1 Introduction

QR (Quick Response) codes are two-dimensional matrix barcodes that are used to encode information. In recent years, they have increasingly found their way into urban spaces and mobile communication to make the Web instantly accessible. The most common use case is to encode a link or other textual information to make it instantly available, obviating the need for a user to type a URL manually. Due to their high information density and robustness, QR codes have gained popularity as part of the *Internet of Things*.

Applications range from simple web links in billboard advertising to monetary transactions during which highly sensitive data is handled. Besides a broad range of advantages, QR code processing on mobile devices often implies distinct security vulnerabilities of which the average user is mostly unaware. Attackers misuse QR codes to encode malicious links that lead e.g. to phishing sites or to the execution of malicious code. These malicious QR codes can be printed on small stickers and pasted over preexisting QR codes on billboard advertisements [182]. The importance of QR code security has been acknowledged by previous research such as Seeburger et al. [165] and Vidas et al. [186]. However, to the best of our knowledge, we are the first to investigate the impact of usability on risk perception with respect to malicious QR codes. As QR codes are increasingly used as an attack vector, there is a need to develop effective tools that help users to protect themselves against such attacks. In this chapter, we provide an extensive study of QR code reader vulnerabilities with an emphasis on usable security measures. In the first phase of this study we focused on the decoder software and in the second phase on the user. During the first phase, we analyzed QR code reader software for mobile devices. We identified the 12 most frequently downloaded QR code reader applications for Android, iOS and Windows Phone and analyzed them with respect to security protection mechanisms and privacy violations. To test these applications, we produced a set of malicious QR codes and observed how the software processed them. Furthermore, we used HTTP(S) interception to monitor the information which was exchanged between the QR code reader application and the related web service. We found that most applications were not only unable to identify malicious QR codes and hence redirected the user to a malicious web site, but also significantly violated the user's privacy by transmitting personal information to third parties. The second part of our survey consists of an intercultural comparative user study. We identified intercultural factors of security awareness with respect to QR code security. To conduct this study, we encoded a link to a short online survey in QR codes and used them to simulate a phishing attack. We deployed 784 QR code stickers in different locations, such as public toilets, bus-stops and cafeterias in four different European capitals, namely Athens (Greece), Helsinki (Finland), Paris (France) and Vienna (Austria). Our findings not only identify the intercultural differences amongst Europeans but also highlight the need for security improvements to make QR code processing a secure user experience. Based on our comprehensive analysis of security and privacy flaws concerning smartphone applications, we propose a set of design recommendations to improve the QR code encoding scheme, the reader software and the app's usability. To evaluate these recommendations, we implemented a prototype application where security, privacy and usability are considered likewise and showed that our recommendations can help to build applications that are both secure and usable.

4.2 Related Work

QR codes in general consist of different areas reserved for specific purposes. Some areas are functional parts of the QR code and cannot be recovered with error correction. The

data is encoded in black and white *modules* [55]. When QR codes are used as an attack vector, they can be either partially modified or completely pasted over. We refer to partial modifications if singular modules are inverted from white to black and vice-versa. In general we distinguish two types of malicious modifications. The first approach includes the modification of black and white pixels, whereas the second one restricts to alterations from white to black. The latter scenario resembles an attacker modifying an existing QR code by using a black pen as described by [107]. Considering the structure of a QR code, the modification of specific modules can influence the code: changing the character encoding forces the QR code scanner to interpret the data differently. The character count indicator reveals the length of the data and a modification thereof tricks the scanner into reading less of the data or into interpreting modules that are not intended to contain data. Furthermore, various encodings can be mixed within the data to change the modes or to insert and delete respective segments. Attackers can also hide malware by misusing control characters [122]. Attacks performed via QR codes are mainly targeted at automated processes and human interactions. According to [166], attacks on automated processes refer to SQL and command injections tricking the system into performing something other than intended by the process owner. Furthermore, a QR code can be misused for browser-based exploits and cross-site scripting attacks [112]. Dabroswki et al. [48] proposed barcode-in-barcode attacks by generating barcode polyglots. In this chapter, we focus on the human factor as the most vulnerable component. Humans are not able to decode a QR code by simply looking at it. Hence usable tools are necessary to support the user in deciding whether a QR code is malicious or benign. The importance of human factors in QR code-based attacks has already been acknowledged by the scientific community. Seeburger et al. [165] investigate how users interacted with QR Codes stickers in urban spaces called *PlaceTagz*. These *PlaceTagz* were deployed in different locations in Melbourne such as cafeterias, libraries and public toilets. Their results suggest that curiosity is the main motive for dwellers to scan non-contextual QR codes. Thus, with curiosity being the major motivation to interact with an unknown source, users are ignoring the security threats associated with QR codes from unverified sources or are unaware of them. Vidas et al. [186] described QR code-initiated phishing attacks by conducting two experiments in the city of Pittsburgh, a *surveillance* and a *QRishing* experiment. Within their surveillance experiment, they observed how users interacted with the code and if they scanned the codes or not. Furthermore, they observed the proportion of users who scanned the code but refused to visit the encoded URL by visually monitoring user interactions with QR codes. To do so, they deployed a poster with a QR code and a camera to record the user interactions. In their so-called *QRishing* experiment, they deployed QR codes on three different types of posters and flyers to assess the susceptibility of such a phishing attack. In their codes, a link to a survey was encoded. This survey contained a set of questions to identify the initiatives and the behavior of the people that scanned the QR codes. Similar to Seeburger et al. [165], Vidas et al. [186] found that curiosity is the main motivation for smartphone users to scan a code. The research conducted by Vidas et al. [186] is significantly related to our research. However, it does not include an intercultural comparison of the user awareness

and focusses more on how people generally interact with QR codes in urban spaces. Yao et al. [195] proposed a solution called *SafeQR* to harden QR code readers with existing APIs as well as a visual warning scheme for Android apps. Based on the findings by Yao et al. we propose our set of guidelines to harden the QR code itself as well as the reader app and to support the user in verifying the originator of the QR code and to differentiate malicious from benign QR codes. In recent years, intercultural factors of privacy perceptions and security awareness have been examined in scientific literature.

4.3 Evaluation of QR Code Readers

Table 4.1: Features and business data of the sample

	Additional Features			Usage		
App Name	Cloud Sync	Price Compare	Local Shop Finder	Price	Est. User Base	Platform
Scan	✓			$1.99	50-100M	Android/iOS/Win
Barcode Scanner				free	50-100M	Android
RedLaser		✓	✓	free	50-80M	Android/iOS/Win
Bakodo		✓		free	30-55M	iOS
QR Droid				free	10-50M	Android
Quick Scan		✓		free	20-40M	iOS
ShopSavvy		✓	✓	free	15-38M	Android/iOS/Win
QR Code Reader and Scanner				free	10-20M	iOS/Win
Qrafter		✓		free	8-16M	iOS
ScanLife		✓	✓	free	5-12M	Android/iOS/Win
i-nigma				free	5-10M	Android/iOS/Win
AT&T Code Scanner				free	5-10M	Android/iOS

For our evaluation, we selected the 12 apps with the largest user base [4, 192]. Many QR code reader do not only decode QR codes, but offer additional features. At first we evaluated the functional range of the applications and summarized the main features in Table 4.1. Then we produced a sample QR code with a link to a website. Additionally, we modified this QR code in a way that the encoded link minimally deviates from the original link, simulating a phishing attack. However, we switched only a small number of modules so that the human eye would not be able to detect the deviation from the original QR code. Then we scanned the codes and observed if the decoder application was able to detect the potential phishing attack and if they verified the trustworthiness of the decoded content. Additionally, we reviewed the APIs published by the application vendors, if available. Finally, we used an HTTP and HTTPS interception proxy to observe the communication between the mobile application and the external web services to determine if sensitive information was leaked through the reader application. In the following, we describe the investigated features in detail.

Sample Description and Features

Table 4.1 provides an overview of the selected applications, their additional features and estimated user base. The additional features we identified within our application sample are as follows: *(1) Synchronization:* indicates that the application synchronizes the scan history across the Web and all registered devices. This feature may have additional benefits for the user, but also yields distinct privacy and security challenges. *(2) Price*

Comparison: After scanning a QR code, the decoded information on a product or service is used for an online price comparison. The application then displays the current best price of the respective product amongst different online shops. This beneficial feature however involves the transmission of sensitive information to a third party and provides this third party with data to track the user. *(3) Local Shop Finder:* In addition to a price comparison, the application finds shops near the user's current and/or requested location that offer the scanned product. This additional feature also requires location information in order to function and enables the third party to track the user.

Security Protection Mechanisms

We evaluated the QR codes in reader applications with respect to basic security enhancements to protect the user against malicious QR codes.The results of our evaluation are summarized in Table 4.2. In the following section, we describe these security features in detail.

(1) Modification Detection: We modified a QR code with a link encoded to the website of the credit card company *American Express* (`https://www.americanexpress.com`) in order to redirect the user to `https://www.aoericanexpsers.com` (attack proposed in [107]). Just like in a real-world phishing example, the URI is modified in a way to look similar to the original URI. We then scanned both QR codes and examined if the QR code reader application detects the modification and notifies the user regarding a potential fraud.

(2) Website Analysis: We investigated if the respective application uses *Google Safe Browsing* [5] or a similar service. Google Safe Browsing is a service that checks URIs and web content if they for malware or phishing content.

(3) URI Display: Displaying the URI is a powerful feature to notify the user about the actual encoded URI and can prevent the user from becoming a victim of a social engineering attack. As QR codes are not human-readable, users depend on this functionality for awareness of the encoded content.

Privacy Violations

To analyze the leakage of personally identifiable information (PII), we analyzed their network communication. To do so, we used an HTTP(S) interception proxy to monitor information exchanged between mobile reader software and their corresponding web services.

(1) External Communication: QR code reader applications process captured images directly on the device. Nevertheless, many applications transmit the decoded content of the QR codes to the servers of the application vendor.

(2) User Tracking: In general, QR code reader applications do not require personal information and user tracking in order fulfill the user's requirements.

(3) Location Data: QR code reader apps do not require the current location of the user in order to function. However, if they provide additional features as shown in Table 4.1, location information may be required.

Results

In the following we present the results of our security evaluation with respect to our evaluation criteria. Table 4.2 shows the results of our evaluation with respect to security enhancements. Fields that are highlighted in red indicate that the (non-) existence of a feature has a negative impact on security, whereas green indicates that the result is beneficial for security. None of the 12 evaluated applications was able to detect a modified QR code. Furthermore, only a single application (QRafter) analyzed decoded URIs with a service that checks URIs and web content for malware and phishing. The majority of the QR code readers within our sample provided a feature to display the decoded URI. The software with the biggest user base in our sample did however not provide this security enhancing mechanism. Some applications provide an option to disable this functionality. Table 4.1 outlines our results with respect to privacy violations. 10 out of 12 applications transmit the decoded QR code content to remote servers. While a number of applications require external communication with an associated web service to provide additional features, four applications (QR Droid, ShopSavy, i-nigma and AT&T Code Scanner) use this external communication to leak user information even though the application does not require it. All evaluated apps do not need any login or any other form of authentication. However, most applications within our sample that transmit the decoded content to external servers generate a unique user identifier to track the user's activity. ShopSavvy for example regularly transmits user information together with the decoded content[1] of the QR code (*content*), the user's current location (*country*) and a generated user identifier (*user*). Quick Scan and Qrafter regularly transmit information to external servers but do not generate a user ID to track the user. Over one third of the tested apps required permission to access the device's location and transmitted location information to a third party, even though some of them do not provide any of the listed additional functionalities. Concerning privacy, only 2 out of 12 surveyed applications (Barcode Scanner, QR Code Reader) were, according to our criteria, sufficiently protecting the user's privacy. The AT&T Code Scanner significantly violates its user's privacy as it scored worst in our privacy as it regularly collects and transmit privacy sensitive information and requires permission to location data. However, as shown in Table 4.1, it does not provide additional functionality that would justify this behavior.

4.4 User Study

In order to determine if there are any significant differences in security awareness, we conducted an intercultural comparative user study. In this section, we outline the construction of our study and present our results.

[1] `https://api.shopsavvy.com/5/cloud/scans/qrCode`

App	Callback	User Tracking	Location Data
Scan	Yes	Yes	No
Barcode Scanner	No	No	No
RedLaser	Yes	Yes	Yes
Bakodo	Yes	Yes	Yes
QR Droid	Yes	Yes	No
Quick Scan	Yes	No	No
ShopSavvy	Yes	Yes	Yes
QR Code Reader	No	No	No
Qrafter	Yes	No	No
ScanLife	Yes	Yes	Yes
i-nigma	Yes	Yes	No
AT&T Code Scanner	Yes	Yes	Yes

Figure 4.1: Privacy violations of QR Code readers

App	Detect Modifications	Analyse Website	Display URI
Scan	No	No	No
Barcode Scanner	No	No	Yes
RedLaser	No	No	Yes
Bakodo	No	No	Yes
QR Droid	No	No	Yes
Quick Scan	No	No	Yes
ShopSavvy	No	No	No
QR Code Reader	No	No	No
Qrafter	No	Yes	Yes
ScanLife	No	No	No
i-nigma	No	No	Yes
AT&T Code Scanner	No	No	No

Figure 4.2: Security features provided by QR code readers

Design and Recruitment

We deployed an online questionnaire through QR codes in four different European countries, namely Austria, Finland, France and Greece. The QR code stickers were placed in highly frequented urban spaces, such as bus stops, public toilets or universities. We did not replace existing QR code stickers. We deployed three different types of QR code stickers: plain QR codes with no additional information, QR codes with a description (respectively translated) and QR codes with cute cat images. The QR codes did not only have the link to the survey encoded, but also a city parameter, a location parameter and a unique ID to measure the performance of each sticker. Our goal was to simulate a QR code phishing attack. In this way, we were able to recruit participants that scan unverified QR codes in public spaces. As our study was conducted in an unobserved manner, it is limited to people who decided to scan the codes. We are therefore unable to determine how many people noticed our codes, but decided not to scan them. However,

this selection bias is often exploited within a real world attack scenario as presented in [84]. In our study, we focus on this group of QR code users as they are the most vulnerable group. In our experiment, the participants came across the QR code just like it would be the case in a real-world scenario. Especially with respect to the QR code stickers without further information, the participants did not know about the encoded content before actually decoding it. After scanning a QR code, the participants were redirected to our website with a seven questions-long questionnaire. The survey was designed in a way that it was easy to handle on a smartphone and could be answered within a few minutes. We translated the survey questions to the languages spoken in the respective countries. The study was conducted simultaneously in four cities. We deployed 784 stickers in total, 113 (14,4%) were utilized by participants at least once (273 hits). The data was collected within two months and 83 participants completed our online survey. As all stickers were placed in locations directly in the city center, it is mostly limited to urban dwellers and excludes participants from rural areas. As one third of the stickers was deployed near university campuses, there is a considerable population bias. We measured the performance of our stickers and evaluated how many people scanned our codes but did not fill out the survey, but did not estimate the fraction of people who saw our stickers but decided not to scan them. Simulating a social engineering attack is an ethically sensitive area. Therefore, we decided not collect any individual-related information except for age and gender. Furthermore, we provided the participants with an opt-out option. For ethical reasons we did not make any false pretenses or impersonate someone else. We also refused to replace already existing QR codes to avoid financial damage or reputation loss of the advertising companies. We followed the guidelines from [163].

Results

In all four cities, the stickers with plain QR codes and no additional information performed best, followed by those with a description. In all four cities, the stickers deployed around universities performed better than stickers near public transportation and toilets. According to the survey results, French and Greek participants scan QR codes more often than Finns and Austrians (Figure 4.3). French participants scan QR codes significantly more often than Finns (Independent two-sample t-test, probability of error $p = 0,043$) and Austrians (Independent two-sample t-test, probability of error $p = 0,03$). The performance measures of our stickers support this statement.

The reasons for scanning the QR codes were also similar amongst the surveyed cultural groups. Most participants scanned the QR codes out of curiosity or boredom (similar to [186] and [165]), regardless of a specific location where it was placed. In Greece, all the participants reported curiosity as the main motivation to scan the codes.

Concerning risk perception, the indicated answers varied among the cultural groups as shown in Figure 4.4. In Paris, more than 70% of the participants perceived our QR codes as fishy or reported to be sceptical towards QR codes. In comparison, less than 30% of the participants from Athens perceived our stickers as risky. This is effect is

statistically significant within our sample (Independent two-sample t-test, probability of error $p = 0,05$).

About half of the French participants stated that they generally suspected a potential threat.

More than 50% of the Austrian, Finnish and Greek participants reported to check the encoded URL before actually visiting it. However, almost the same fraction of our sample used QR code readers which do not provide this option.

Our sample shows a significant gender imbalance between the four European cities. 88% of the entire sample reported being male. Figure 4.5 illustrates the gender distribution. 55% of all the participants reported being between 18 and 24 years old. Around 25% were between 25 and 30 years old. The oldest participant was 45 years old and participated in Vienna. Our results suggest that young men are the largest user group of QR codes. These findings correspond to results published in [131].

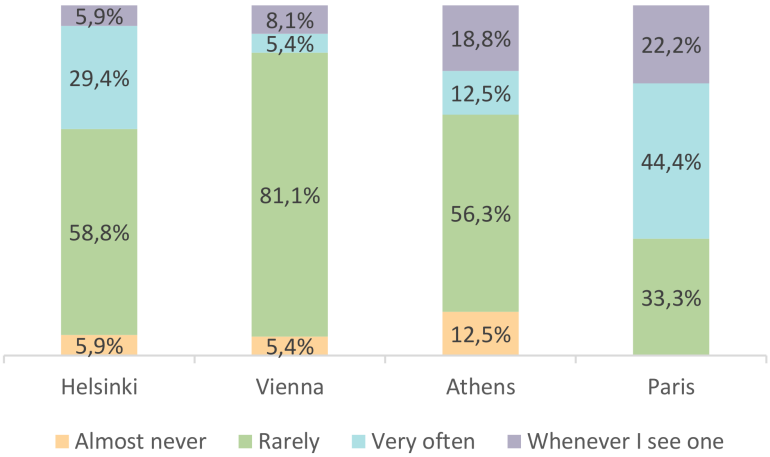


Figure 4.3: Self-reported QR code scanning frequency in percent. $n_H = 17$, $n_V = 37$, $n_A = 17$, $n_P = 12$

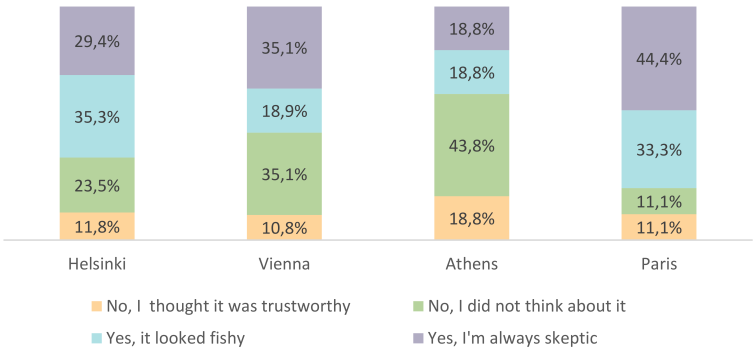


Figure 4.4: Self-reported considerations on the trustworthiness of our QR code stickers. $n_H = 17$, $n_V = 37$, $n_A = 17$, $n_P = 12$

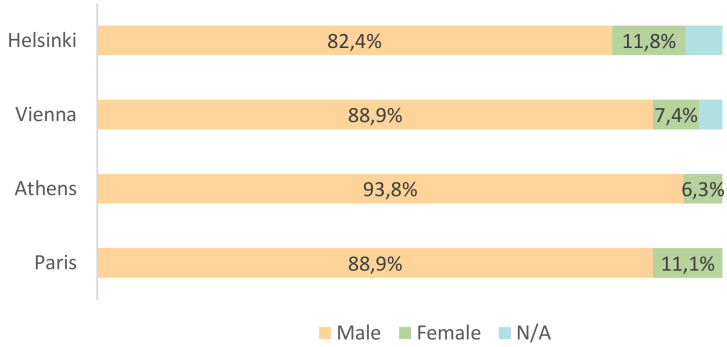


Figure 4.5: Gender distribution by city in percent.

The most popular operating system within our sample was Android. 70% of the Greek participants accessed our website with a device running Android. The second most popular operating system was iOS. However, almost one quarter of participants from Helsinki were using Windows Phone, which can be explained by the relatively high market share (23%) of Nokia Lumia phones in Finland [7].

4.5 Discussion

In the course of this evaluation, we found that security enhancing features are very rarely provided by QR code reader applications. Considering the fact that malicious QR codes are a potential threat [182] it is surprising that most QR code reader applications ignore this danger. Services, such as Google Safe Browsing, anti-phishing or malicious URI detection tools (e.g. [127]) provide effective security measure. Even though these tools are mostly free and easy to use, 11 out of 12 applications do not use them. Due to the fact that QR codes are not human-readable, the user relies on effective tools to identify whether the QR code is malicious or benign. Therefore, we consider URI display as essential to support the user's decision making process on whether to visit the encoded link or not. Therefore, we consider applications that do not display the URI as containing a major security weakness. Our results show that many applications leak user-related information. However, some of them require the transmission of this information to external web services in order to provide full functionality. Nevertheless, 4 out of 12 applications leaked information without any functional benefit for the user. In general, the results of our intercultural user study suggest that Finnish QR code users have the highest level of security awareness and successfully apply their knowledge in a real world scenario. When comparing our results to those published by Seeburger et al. [165] and Vidas et al. [186] the overall attitude towards QR codes as a medium to access Web content is comparable for European, American and Australian smartphone users. However, the awareness of security-related challenges varies, even among European nations. Due to the different perceptions of privacy in non-western countries, we plan to expand our survey to not only a more in-depth analysis of the European data, but also Asian countries such as Japan and China, where the overall performance of QR codes

in public spaces is significantly higher. This is due to the fact that in comparison to one-dimensional barcodes, two-dimensional barcodes have the ability to encode complex Chinese characters. As our results suggest, young men are the most considerable user group of QR codes. This gender imbalance motivates more gender-specific research concerning awareness, technology acceptance and software design. We showed that our design recommendations can, if considered in the architectural design decisions of an application architecture, effectively protect users from QR code-based attacks. In particular, the findings from the usability study were very helpful in designing effective warnings and a user interface that supports the user in the decision-making process on whether to trust or not to trust a certain source of information. Our findings also highlight the importance of user-centric design approaches for secure applications.

4.6 Conclusion

In this chapter, we presented an extensive analysis of QR code security from a holistic point of view. We analyzed why users are still susceptible to QR code based attacks and why currently deployed smartphone applications are unable to mitigate these attacks. Additionally, we proposed a set of design guidelines to build secure, usable and privacy-preserving apps. We showed that if security, privacy and usability aspects are considered evenly, smartphone applications can effectively protect users against malicious QR codes.

CHAPTER 5

Privacy in the Age of Wearable Computing

In the coming age of wearable computing, devices such as Google Glass will become as ubiquitous as smartphones. Their foreseeable deployment in public spaces will cause distinct implications on the privacy of people recorded by these devices. Particularly the discreet recording capabilities of such devices pose new challenges to consensual image disclosure. Therefore, new *Privacy Enhancing Technologies (PETs)* will be needed to help preserve our digital privacy. At the time of writing, no such PETs are available on the market to communicate privacy preferences towards Glass. In the scientific literature, a handful of approaches has been presented. However, none of them has been evaluated regarding their affordances and overall usefulness. In this paper, we provide the first systematization and qualitative evaluation of state of the art PETs that were designed to communicate privacy preferences towards (wearable) cameras, such as Google Glass. In addition, we present three conceptual PETs derived from scientific literature and a user study to evaluate the feasibility of these concepts. For the user study, we conducted semi-structured interviews in public spaces. Our results suggest that users prefer a wearable PET which works regardless of context and is easy to operate by simply pushing a button.

This chapter is an extended version of [110] and [111].

The remainder of this chapter is structured as follows: Section 5.1 provides an introduction and 5.2 presents related work. Then we present a systematization of wearable PETs in 5.4 and a qualitative user study in 5.5. Section 5.6 provides a discussion and Section 5.7 the limitations of this chapter. Section 5.8 concludes this chapter.

5.1 Introduction

Wearable computers with integrated cameras such as Google Glass might soon become as ubiquitous as smartphones. Due to their hands-free user interface and the discreet recording capabilities, collecting and sharing images and videos becomes easier than ever. In contrary to smartphones and other mobile devices, Google Glass literally remains in the wearer's face all the time. Consequentially, many bystanders view such wearables as invasive and fear substantial implications on their digital privacy. Since the paradigm shift to user-generated content on the Internet, the awareness for picture privacy has risen. The foreseeable deployment of wearable technology in public spaces is about to multiply the set of challenges related to non-consensual disclosure of graphical material on the Internet. In such situations, getting informed consent of all people recorded by such a device is infeasible. Recently, attacks against Google Glass wearers in public have been reported in the media [2]. These scenarios highlight the high societal demand for *Privacy Enhancing Technologies (PETs)*. At the time of writing, no PETs are available on the market to communicate privacy preferences towards wearable cameras. In scientific literature, a handful of approaches has been published, however none of them placed an emphasis on whether they are actually useful if deployed in particular situations where users are constrained in what artifacts they can carry or wear. In this chapter we provide a first systematization of PETs that have been published in scholarly articles. To do so, we propose a collection of properties and criteria for categorization. Our systematization provides a first suggestion how a standardized evaluation framework for (wearable) PETs could look like. In the course of our extensive literature review, we found approaches of how future PETs might look like. Most of them however, have substantial limitations such as that they address a narrow scenario, exclude particular user groups or cause further privacy challenges due to their privacy-violating functionality. These limitations may reduce the user's subjective satisfaction and introduce errors. Hence, they potentially have an impact on user experience. Additionally, we explore how privacy preferences can be communicated towards disruptive cameras in privacy-sensitive spaces such as public beaches, where users are constrained in what technology they can carry and use. In order to get an informed consent between photographers and bystanders, we designed three conceptual privacy-mediating technologies based on related literature: a smartphone app, a privacy-bracelet and a clothing-based approach. We then conducted 20 qualitative interviews to study peoples' privacy feelings towards disruptive cameras at a beach and in a cafe and their attitudes towards our approaches. We found that there is high demand for such tools irrespective of location and that a dedicated privacy device was preferred by most of the participants. Denning et al. [54] presented a study on individuals' reactions when they are bystanders around lifelogging devices with first-person cameras.

We conducted 20 semi-structured interviews in the field. We conducted interviews at a local beach, where we approached people wearing bathing wear as well as in a local cafe. Our goal on the one hand was to gain an understanding of users' privacy concerns towards wearable cameras such as Google Glass, and on the other hand which kind of PET they would like to use to preserve their privacy. We were particularly interested in whether the

location of the interviews would influence the participants' perceptions and preferences. Our hypothesis was that in the beach environment the clothing-based PET would be the preferred option, while in the cafe environment there would be a mix of preferences. Our findings show that most of our participants had serious concerns regarding their privacy when confronted with Google Glass. Furthermore, and to our surprise, we found that the privacy-bracelet was the preferred PET for most of our participants, irrespective of location.

5.2 Related Work

Various methods have been proposed to allow individuals to defend their privacy against non-consensual disclosure of pictures and videos. As concealing one's face (e.g. with a mask) is not socially or legally accepted everywhere, several methods have been proposed to communicate picture privacy preferences towards cameras.

The respectful cameras approach as presented in [161] uses hats and scarfs as visual markers. The picture privacy policy framework presented in [49] uses a similar approach. Contrary to [161], the picture privacy policy framework uses not only accessories but also T-shirts to encode privacy policies. The used encoding scheme is designed in an unobtrusive way with almost no impact on apparel appearance. FaceBlock [196] uses biometric features as visual markers instead of wearable artifacts.

As most portable devices come with GPS sensors, location-based technologies such as the SnapMe privacy watchdog [83] or Blind Spot [147] are feasible to mediate privacy preferences. In comparison to SnapMe, the Blind Spot approach is based on fixed cameras and intended for CCTV-like surveillance systems. Halderman et al. [76] presented a location-based privacy management protocol. Barhm et al. [23] presented a an approach where individuals perform gestures when recorded by a camera to be made irrecognizable.

In this work, we focused on concepts that enable bystanders to control their privacy in different situations. In comparison to these concepts, PlaceAvoider [181] was designed to blacklist specific locations instead of individuals. Also, the control does not lie in the hand of the bystander. ScreenAvoider [108] was designed to protect sensitive computer screens instead of individuals. Similarily, PrivateEye [155] was proposed to protect sensitive content. WaveOff [155] could potentially be used to protect persons in public spaces. In contrary to our approaches however, the wearer of a lifelogging device controls the privacy options instead of the bystander. The work by Hoyle et al. [91] provides insights in how wearer of a lifelogging device perceive privacy in a lifelogging context and focuses on the wearer's perspective (it also covers how wearers perceive bystander reactions) and found that p̈eople may prefer to manage privacy through in situ, physical control of image collections.̈ Roesner et al.'s [159] approach relies on a centralized authority and compared to our approaches allows to specify policies for users and objects irrespective of context.

5.3 Systematization of PETs

5.3.1 Properties

The properties presented in this section haven been selected as a set we believe highlights important evaluation dimensions with respect to usability, in particular subjective user satisfaction, learnability, memorability, errors and efficiency.

User-Initiated: In order to mediate privacy-preferences, some PETs require a user to perform an action. Due to the unobtrusive recording capabilities of wearable cameras, users may be recorded by such a device without actually being aware of it. Therefore, user-triggered mediation hinders the consequent communication of a user's privacy preference. This is very likely to cause errors and misunderstandings which are very likely to have a negative impact on the overall user experience.

Location-Based: As determined by Denning et al. [54], privacy preferences can be determined by certain situations or locations. The definition of a privacy-sensitive space mostly varies from user to user, highly depending on their socio-cultural background. However, most users may require a device that works regardless of a certain location. Therefore, location-dependency may be a limiting factor. As a location-based approach usually requires the transmission of location information to other entities (e.g. a trusted server via a secure channel or another device in the surrounding), new privacy challenges are implied.

Face-Recognition-Based: In order to correlate the user of a PET with a user in an image or video taken by a wearable camera, facial recognition could be an efficient method of choice as state of the art algorithms provide sufficient accuracy to correctly identify individuals. Certainly, facial recognition requires the transmission of privacy-sensitive data to an associated service and therefore poses significant challenges to preserve the user's privacy.

Visual-Marker-Based: As widely used in augmented reality applications, person-tracking can efficiently be performed using visual markers. For PETs, this means that a user has to carry or wear one or multiple visual markers in order to communicate privacy policies towards wearable cameras. Markers can be designed in an obtrusive or unobtrusive way. In certain situations or cultural groups, subtle markers could be preferred over invasive ones and vice versa.

Gesture-Based: In augmented reality applications, gestures are another feasible approach to track individuals in videos. This approach is mostly limited to long or full-shot videos as it is obviously difficult to perform gesture-recognition on single images or smaller image selections. Furthermore, gestures have to be actively performed and therefore require a user who is aware of being filmed. It furthermore poses distinct accessibility challenges for people with physical disabilities and elderly users.

Signal-Emission-Based: As cameras are sensitive to a certain light spectrum, signal-emitting jamming could be used in PETs. Signal-emission based approaches are mostly

expensive as they require a dedicated technical artifact which makes them significantly obtrusive.

Physical-Artifact-Required and/or Dedicated-Device-Required: This is an umbrella property for all approaches that require a dedicated physical artifact in order for the PET to function (e.g. an electronic device or visual markers). From the user's perspective, tangible interfaces as provided by such artifacts offer advantages and disadvantages. The main disadvantage is that the user has to carry or wear the artifact at any time. In some situations, this is unfeasible (e.g. in spaces where digital artifacts cannot be operated due to environmental constraints). In contrary, physical artifacts are often easier to understand for the user and give them a sense of control.

Requires-Trusted-Third-Party-Service: If communication with an external service is required, an Internet connection is indispensable. This might be unfeasible in some scenarios, where users are limited in what devices they can carry (e.g. at a beach)

Smartphone-Based: Smartphone-based approaches are easy to deploy since most individuals in today's society carry one with them all the time. However, situations where smartphones are not applicable but preserving an individual's privacy is required. An example of such a situation would be sunbathing and wearing only a bikini.

Visibility: Some PETs require physical artifacts in order to function. When deployed, some of them are highly visible to bystanders and therefore instantly disclose a certain privacy preference to nearby individuals. In some cultures or even particular situations, a more subtle and unobtrusive technology could be preferred by the user. However, others may want to use an obtrusive PET in order to disclose their privacy preference or policy openly when recorded by a wearer of Google Glass or a similar device. Low visibility however may constrain the communication of a cognitive model towards the user. It also implies a lack of feedback options.

Accessibility: As digital privacy affects all user groups likewise, a PET should work regardless of disabilities or other physical or cognitive conditions, such as low motor control or visual impairments.

Anonymity: PETs should not imply further privacy violations due to their functionality. Approaches that use facial recognition or location-tracking potentially violate the privacy of their users. Presumably, users of PETs are highly concerned about their privacy and potentially perceive privacy-violating PETs as paradoxical.

Impacts-User-Behavior: Some PETs heavily impact the user's behavior, as they either require a high effort in preparation or require the user to perform an action. PETs that require a user to be aware of being filmed also potentially influence the user's behavior.

Requires-Devices-To-Comply: This property indicates whether a PET can be deployed only if (wearable) cameras are updated accordingly (software and/or hardware).

5.3.2 Evaluation

Table 5.1 presents our systematization in which we indicate if a certain PET has a certain attribute or not. If a PET could be configured in a way to evoke a certain property, we assume a best-case working scenario supposing that a poor implementation would make any concept potentially unusable. Obviously, some properties are disadvantageous for the overall user experience. For this systematization, we refrain from introducing a rating scheme and prefer a non-judgmental presentation. The reason for this is that neither we nor the authors of the respective PETs conducted user studies that would confirm these assumptions. While some of the PETs presented in this section have been particularly designed for the mobile/wearable computing domain, others were designed to preserve picture privacy in general. For the purpose of fostering a fruitful discourse however, we discuss some potentially impacting factors in a qualitative way.

The **Privacy Makeup** and hair-style approach as presented by Harvey et al. [81] exploits the weaknesses of commonly used face detection systems. To inhibit the feature response of face detection algorithms, significantly invasive distortions are created with camouflage makeup. This approach is time consuming in preparation and visually dominant. It therefore hinders everyday social interaction and can provoke unwanted reactions. It is only feasible when facial recognition algorithms are used or the makeup is applied in a way that its wearer is unrecognizable. For unexperienced users, it is hard to apply

Table 5.1: Systematization of PETs

	User-Initiated	Location-Based	Face-Recognition-Based	Visual-Marker-Based	Gesture-Based	Signal-Emission-Based	Dedicated-Device-Required	Physical-Artifact-Required	Communicates-With-Server	Internet-Connection-Required	Smartphone-Based	Visibility	Accessibility	Anonymity	Impacts-User-Behavior	Requires-Devices-To-Comply
Privacy Makeup [81]				●								●		●	●	
Respectful Cameras [161]				●				●				●	●	●		●
P3F [49]				●				●								●
OfflineTags [145]	●			●				●				●	●	●	●	●
Privacy Visor [193, 194]						●	●					●	●	●		
SnapMe [83]		●	●						●	●	●					●
FaceBlock [196]		●	●						●	●	●					●
BlindSpot [147]		●				●	●						●	●		●
Place Avoider [181]		●											●	●		●
Privacy Gestures [23]	●				●							●		●	●	●

the makeup correctly. The **Respectful Cameras** approach as presented in [161] uses colored hats and scarfs as visual markers. Depending on whether an individual prefers to be made irrecognizable or not, the corresponding artifact is chosen and worn in front of a camera. The Picture-Privacy-Policy framework (**P3F**) as presented in [49] uses a similar approach, however the privacy policies used in this scheme are more complex and fine-grained. The visual markers of the respectful cameras approach [161] is based on a binary privacy policy and obtrusive markers. The P3F use not only dedicated accessories but aims at providing a clothing pattern database with fashionable clothing patterns that are then used as visual markers. A large-scale deployment as presented in the chapter, however, would require all cameras or picture publishing platforms to use the P3F software to detect the visual markers and to deduct the privacy policies from them. Another visual-marker based approach is **Offlinetags** [145]. Offlinetags uses four different symbols readable by the open-source Offlinetags software. These symbols can simply be printed on a piece of paper and then presented to a camera. In contrary to the other visual-marker-based approaches presented in this section, the obtrusive markers must be presented actively towards a camera. Yamada et al. [193, 194] presented the **Privacy Visor**, i.e. glasses with infrared light sources that are visible to most camera sensors but invisible to the human eye. The goggles approach requires a constant power supply and infrared LEDs that can keep up with the ambient light. As most portable devices come with GPS sensors, location-based technologies such as the **SnapMe** privacy watchdog [83] or **Blind Spot** [147] are feasible to mediate privacy preferences. These approaches are based on correlated location information of a camera and its bystanders. Additionally to the location-reference, SnapMe proposes the use of facial recognition to identify individuals in pictures. In comparison to SnapMe, the Blind Spot approach is based on fixed cameras and intended for CCTV-like surveillance systems and thus limited to a specific location. **FaceBlock** [196] is based on biometric features on images taken by a (wearable) camera. Similar to the other facial-recognition-based approaches in this section, the FaceBlock system implies further privacy challenges, as privacy-sensitive biometric information is processed and transferred to a (trusted) server. Both FaceBlock and SnapMe provide a smartphone app where users can configure their privacy-settings. The **PlaceAvoider** [181] approach is not only intended to protect the privacy of bystander but also of the wearer of a wearable camera. Similar to BlindSpot, it provides blacklisting of privacy-sensitive spaces like bathrooms and bedrooms. Similar to other location-based approaches, it requires a predefined location and might therefore not be applicable in all desired situations.
Barhm et al. [23] presented a gesture-based method (**Privacy Gestures**) to communicate privacy preferences. Individuals perform defined gestures when recorded by a camera. Even though no additional artifact is required, its feasibility is limited to situations where an individual is aware of being recorded.

5.4 Conceptual Wearable Privacy Enhancing Technologies

At the time of writing, there is no technical solution available on the market to communicate privacy preferences towards wearable cameras. In scholarly articles, very little attention has been paid to the challenge of designing usable technologies to tackle this issue. In this section, we present three abstract PETs to study users perceptions and attitudes towards these different methods. They have been assembled based on existing approaches and related work as presented above. For the purpose of this study, we presented best-case working scenarios, since we were mainly interested in the attitudes and perceptions of the users to the potential of the concepts. Therefore, we left out many of the technical challenges which still need to be overcome. As shown in previous work, privacy preferences are highly context-dependent [91, 99, 148].

5.4.1 The Privacy App

The *Privacy App* is mainly inspired by the SnapMe [83] and FaceBlock [196] apps. Both apps have a range of configuration options. For the purpose of this study, we defined that the location of the app user and the location of the nearby cameras are transmitted to a photo sharing server together with the privacy preferences of the user. Due to the co-location information the photo sharing service can blur the faces of people with corresponding app configurations when a photo is uploaded. This feature is additionally supported by face recognition software. This concept represents the traditional technology approach.

5.4.2 The Privacy Fabric

The *Privacy Fabric* is a piece of cloth to communicate a user-defined privacy policy. The concept is inspired by P3F [49] and privacy hats and scarfs by Schiff et al. [161]. It is based on pattern recognition and works without additional hardware. To create a privacy cloth, e.g., swimming trunks, T-shirts or any other piece of clothing with a privacy pattern, clothing and accessory manufacturers can use a specific encoder to create a visual marking or pattern that matches any wardrobe style. Either the wearable doing the recording or the photo sharing service can detect if a person is wearing a piece of clothing with a privacy preference encoded in it and can blur those peoples' faces. The main advantage of this method is that it is unobtrusive as no piece of technology needs to be operated. This concept represents the most "wearable" PET and we hypothesized that users would prefer this in the beach scenario since it would allow them to express their privacy preferences in an unobtrusive way.

5.4.3 The Privacy Bracelet

We designed the concept of the *Privacy Bracelet* as a mix between the privacy smartphone app and the privacy fabric. While it uses technology of similar power to the smartphone, it is wearable, similar to fitness trackers (e.g. FitBit[1]). This concept was not based on related work but was designed to give us a half-way point between the two PETs described above and allow us to have middle ground during the interviews to be able to contrast between the two technologies described above. In our concept the privacy bracelet has a simple button to turn privacy on and off. If the privacy button is turned on, the device emits a signal that wearable cameras would be able to detect and blur the faces of the bracelet wearers.

5.5 User Study

5.5.1 Methodology

The aim of this study is to evaluate users' attitudes toward wearables in the two scenarios (beach and cafe) and the PETs concepts presented above. We conducted field sessions with semi-structured interviews at a public beach and in a cafe. 20 participants were recruited. The participants were compensated for their time with ice cream. During the field sessions, two researchers were present.

The interview sessions proceeded as follows: the two researchers approached potential users of the proposed PETs, i.e. groups of bathers wearing bikinis or swimming trunks and people in a cafe. Furthermore, the participants could use Glass with the video and picture capabilities enabled to gather hands on experience.

The interview consisted of two parts: In the first part, we examined privacy concerns related to Glass. The second part focused on the proposed artifacts to express privacy preferences. After a brief introduction to the purpose of the study, the interviews began with questions on Google Glass and privacy. Afterwards, the researchers presented the three PETs as described above. The three methods were described using illustrations in a neutral way without any hints on who developed the method. To preserve the participants' privacy during the recruitment and the interview sessions, all recording functions of Glass were disabled. We recorded the interview sessions (audio only) after the participants gave their consent. Additionally, one of the researchers took notes during and after the interview sessions. The only personal information we collected were age, gender and profession.

The interview questions can be found in the Appendix 9.

5.5.2 Coding

After the data collection, we went through the interviews and produced an initial set of codes. To do so, we traversed the data segments from the interview to each question.

[1] `www.fitbit.com`, last accessed 9/9/2015

Two researchers performed the initial coding independently of each other to minimize the susceptibility of biased interpretation. After the initial coding process, we discussed the retrieved codes, recurring themes, patterns and connections. Additionally, we compared the codes with the ones presented in [54]. After agreeing on a set of codes, we used the codebook for a final coding of the interview data. All interview segments were coded, regardless if they emerged directly from a question or a subsequent discussion.

5.5.3 Results

In this section, we present the results of our user study. As our evaluation is based on qualitative data, we place the emphasis on an exploration of the ideas and insights of the participants instead of a quantitative analysis. However, some of the numbers are given as a rough indicator of trends which we spotted during the study. These will however need to be backed up by a larger quantitative study.
In total, we interviewed 20 participants. They were recruited at a public beach and in a cafe. 9 participants were male and 11 were female, and the age ranged between 19 and 42 (median age: 25). After 20 participants, we reached saturation and little to no further insights were gained, so we concluded the study.

Technology Familiarity

All of the participants had at least a rough idea of Google Glass and its basic functionality. They were all aware of Glass' ability to record pictures and videos. Most of them (17/20) immediately associated a camera with the device when they saw us passing by with it. We collected information on profession and highest level of education. None of our participants was working in an IT-related field. 10 had completed high-school, 4 had a bachelor degree, 2 a master degree and 4 did not complete high-school. To tie our results to existing literature, we based questions on privacy on those from Denning et al.[54].

Privacy Considerations

About 12 participants expressed discomfort and irritation as bystanders of Glass-like devices. They were concerned about their privacy and perceived that they lose control over their images and videos. About half of the participants found it disturbing that they cannot see if the Glass-wearer in front of them actually records a video or not. Six participants even expressed vexation and had serious concerns regarding mass surveillance.

> *"If someone wore it [Google Glass] in front of me, I'd definitely ask him to take it off."*
> P13 (25, male)

> *"I have the feeling that [with Google Glass] something serious is going on concerning surveillance. Maybe Glass performs face recognition in the background and transmits the information about the recorded people to the NSA.*

This would make every Glass-wearer an unintended little helper of the NSA."
P19 (42, male)

In contrast, 8 (younger) participants reported a neutral feeling towards augmented reality devices and continuous recording. Most of them said that, over time, they have gotten used to it and perceive the numerous cameras they are surrounded with as a part of their everyday lives. Remarkably, they did not distinguish between governmental surveillance, CCTV or consumer devices such as smartphones and wearables. One third of the participants said that their privacy concerns vary depending on the context.

PETs Preferences

In general, all participants expressed a strong interest in a privacy enhancing or mediating technology to communicate their privacy preferences towards Glass users. On average, our participants indicated an interest of 4.3 on a 5-point Likert scale, where 1 means no or little interest and 5 means high interest. The lowest indicated number was 3. After presenting the three methods (as described above), 13 participants preferred the privacy bracelet. 4 preferred the app and only 2 the privacy fabric. One participant said that he finds all of the suggested methods useless. We saw no trend difference between participants preferences based on the location we conducted the interview in. We found these results somewhat surprising. We had expected a trend towards favoring the privacy fabric in the beach environment and more of a mix or potentially a trend towards the more traditional smartphone app in the cafe scenario.

The main reason for the preference as indicated by the interviewees was ease of use and convenience. Many who supported the bracelet said that they found the user interface very intuitive. Some also favored the anonymous aspect of how the data is transmitted to the camera. Many participants mentioned that they do not want facial recognition and location tracking, as performed by the privacy app in the background. They perceived the use of such methods in privacy tools as paradoxical.

"The server behind the app bothers me just as much as Google Glass does."
P15 (24 years, male)

Thirteen interviewees favored the privacy bracelet because it does not exclude social network deniers or people without a smartphone. Eight participants also liked the idea behind the privacy fabric but mentioned concerns with respect to personal styling preferences and mentioned it could be complicated to adjust their clothing based on their context-related privacy preferences.

Privacy & Context

As previously stated, one third of our participants indicated that their privacy concerns vary depending on the context. They mentioned parties with alcohol and their own home

as privacy-sensitive spaces. Concerning PETs however, 16 participants said that they made their choice independently from the location of use. They said that such a tool should work regardless of the environment. We observed no qualitative difference in the responses between the groups we interviewed at the beach and the cafe.

> *"In general, I don't really care about privacy. But I would not want to be filmed drinking during a party."*
> P17 (19, female)

Price

Concerning the price people thought the PETs should cost, the suggestions varied greatly. For the privacy bracelet, the lowest suggested price was 10 euros. Three participants said that they would be willing to pay about 150-200 euros. They explained their suggestions based on how highly they value privacy. Many participants said that such a device should not be too expensive so that anyone could afford it. For the app, the highest nominated price was 2 euros. Assuming that the price for a privacy fabric and an ordinary one would be the same, most participants reported that they would buy the privacy fabric due to its additional functionality.

5.6 Discussion

Our results have shown that potential users of PETs want an easy-to-use user interface. For many of them, pushing a button instead of wearing dedicated artifacts is more intuitive and gives them a sense of control. Our results suggest that more and more people desire solutions that work independently of other systems such as smartphones, social networks and other online services that require registration.

To our surprise, our participants showed little trust in the privacy-fabric. The concept was hard to understand and imagine for most of our participants. Therefore, they showed little trust in this method in comparison to the privacy bracelet. This has significant implications for our research, since we had thought this was a very promising novel PET. We also found that the preference for a certain meta-PET did not depend on the location of the interview.

Our results indicate that many users prefer technologies that do not require facial recognition, location tracking and the transmission of sensitive information to (trusted) servers as they perceive this as a violation of their privacy. Furthermore, we found that the preference of a certain PET does not depend on the location. These results pose some serious challenges for the development of future PETs to help mediate privacy preferences in the age of wearable computing.

We chose the two scenarios because we felt they offered good extremes to begin researching the question of how different classes of PETs are perceived in different scenarios. Again since this is a Note we did not want to cover the entire design space but offer insights into specific scenarios that can serve as a starting and calibration point for more broad work. Also, we chose the beach scenario as we wanted to cover a situation in which participants are potentially constrained in what technical artefacts they can carry or wear. The beach is a challenging environment for PETs and to the best of our knowledge has not been studied in relation to PETs yet. We also expected the beach to elicit stronger privacy concern. We find it an interesting result that this did not seem to be the case for our participants. While many other environments are interesting and worth studying we think these two offered a good start and useful insights.

During the interviews, some participants wanted to wear Google Glass and play around with it. To explore its functionality, they had to turn it on and some of them took pictures and videos. After we continued with the interviews, we again disabled all the recording functions. We observed that during the time the participants wore it, most of their concerns vanished but immediately returned when they gave it back to us.

5.7 Limitations

For our interviews, we deployed Google Glass in a public space to confront potential participants with this new technology and to provoke the privacy concerns implied by its presence. As described above, we systematically recruited participants who showed reactions towards Glass for our interviews. Amongst them, most individuals who were 30 or younger immediately agreed to give an interview and showed high interest in this topic. In contrast, many people over 30 refused to talk to us and expressed annoyance and irritation. Since it was significantly harder to recruit participants over the age of 30, and all interviews were conducted in an urban area, the results will probably differ for other demographics. Also, as we conducted semi-structured interviews, we collected self-reported data and as a consequence, our results are based on subjective views and perceptions.

5.8 Conclusion

In this chapter, we presented three different abstract PETs to enhance the privacy of individuals in relation to Google Glass or similar wearables. In 20 semi-structured interviews conducted at a public beach and in a cafe, we examined people's privacy considerations related to Google Glass-wearers in their surrounding.

We found that many people have serious concerns regarding potential privacy violations and that there is high demand for usable PETs. In the course of our interviews, we presented three abstract PETs and asked the participants about their preferences concerning them. Most participants preferred the privacy bracelet, a wearable artifact

with an intuitive user interface that does not transmit sensitive information to third parties. We saw no differences based on the location of the interview. Furthermore, we determined that people prefer a solution that does not exclude particular user groups such as smartphone and social network abstainers.

These results pose significant challenges to future PETs designs since many features our participants found critical are currently used in PETs found in related work.

CHAPTER 6

User Authentication with Force-PINs

Modern, off-the-shelf smartphones provide a rich set of possible touchscreen interactions, but knowledge-based authentication schemes still rely on simple digit or character input. Previous studies examined the shortcomings of such schemes based on unlock patterns, PINs, and passcodes.

In this chapter, we propose to integrate pressure-sensitive touchscreen interactions into knowledge-based authentication schemes. By adding a (practically) invisible, pressure-sensitive component, users can select stronger PINs that are harder to observe for a shoulder surfer. We conducted a within-subjects design lab study (n = 50) to compare our approach termed *force-PINs* with standard four-digit and six-digit PINs regarding their usability performance and a comprehensive security evaluation. In addition, we conducted a field study that demonstrated lower authentication overhead. Finally, we found that force-PINs let users select higher entropy PINs that are more resilient to shoulder surfing attacks with minimal impact on the usability performance.

This chapter is an extended version of a conference paper [116].

The remainder of this chapter is structured as follows: Section 6.1 provides an introduction and in Section 8.2, we discuss related work. In Section 6.3, we introduce the attacker model, the concept of force-PINs, and describe the objectives of this work. Section 6.4 presents the design and results of our lab study. In Section 8.4.1, we provide a security evalution and in Section 6.6.3, we present the results of a field study to show learning effects of force-PINs deployed in a real-world environment. Section 8.6 discusses our work and its limitations and we conclude this work in Section 8.7.

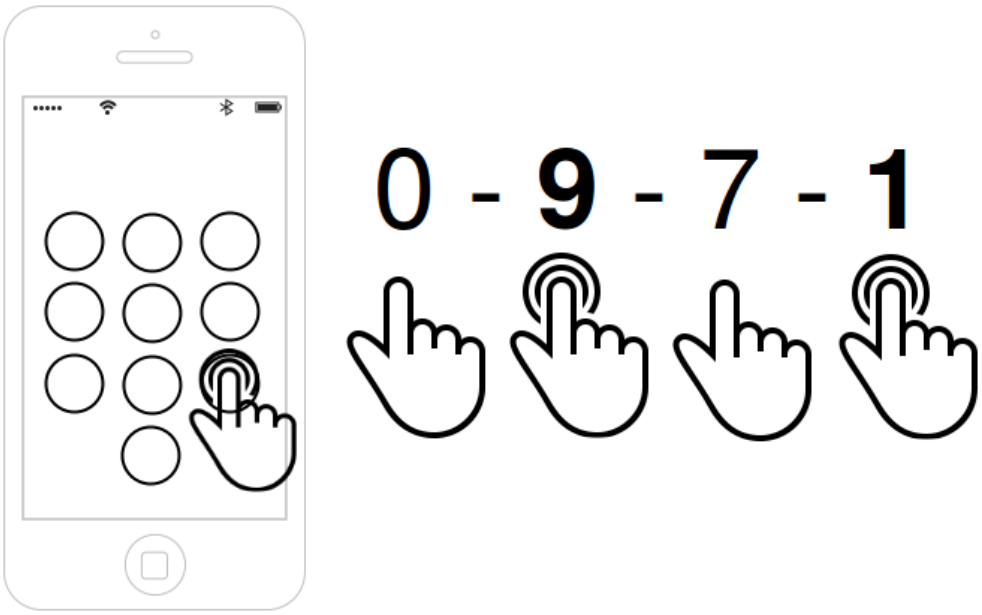


Figure 6.1: Schematic overview of force-PINs: digits can either be entered with shallow or deep pressure on a pressure-sensitive touchscreen, enhancing the space of four-digit PINs to $20^4 = 160,000$ by an invisible component. The user receives vibration feedback as soon as deep pressure is recognized.

6.1 Introduction

With the introduction of pressure-sensitive touchscreens (e. g., Apple recently introduced *3D Touch*[1]), many new kinds of user interaction for smartphones become possible that could also be used to enhance existing authentication schemes. The scientific community has already examined the shortcomings of unlock patterns, PINs and passcodes [79, 52, 18, 172] and presented alternative authentication schemes.

However, none of the proposed systems has shown to be capable of replacing passcodes and unlock patterns as means of authentication. On the one hand, many approaches, e.g., [51, 53] rely on customized hardware that is not available off the shelf and thus makes large-scale deployment infeasible. On the other hand, many alternative approaches, e.g., [45, 121] are time-consuming and therefore increase the authentication overhead. As shown by Harbach et al. [79] in a field study on smartphone unlocking behavior, (un-)locking smartphones produces a significant task overhead. This highlights the need for novel authentication methods that perform equally fast as or even faster than currently deployed systems in terms of authentication speed.

Recently, biometric approaches such as fingerprint sensors and face recognition have found their way into the mobile ecosystem. As with previous authentication methods,

[1] `https://developer.apple.com/ios/3d-touch/`

however, they have shown to be easy to break by attackers and difficult to use for certain groups of users. For example, Apple's fingerprint sensor as found in some recent iPhone models was soon hacked after being introduced [42] and excludes users with weak fingerprints (e. g., due to manual labor). Furthermore, classic biometric methods and implicit authentication based on user behavior still require users to use a PIN for fallback authentication in case the primary authentication methods fail. Bonneau et al. [34] presented a benchmark to evaluate authentication schemes. Their evaluation shows that many schemes only offer minor improvements over passwords (if any) and that many systems offer a number of benefits in theory but show severe limitations in practice. These observations highlight that it is still worth focusing on improving knowledge-based authentication on smartphones as no other authentication method has proven to be as secure and usable as passwords.

In this chapter, we propose that device manufacturers integrate pressure-sensitive touchscreen interactions available on mobile and wearable devices into knowledge-based authentication schemes. Our goal is to improve PIN security by enhancing the password space without compromising usability factors such as authentication time, error rate and memorability. This approach enhances traditional four-digit or six-digit PINs with tactile features using pressure-sensitive touchscreens as found in modern consumer hardware. We refer to these enhanced PINs as *force-PINs* and Figure 6.1 provides an overview of the proposed scheme.

In theory, force-PINs offer the benefit of a larger PIN space by design. Hence they are more difficult for an attacker to guess and are more resilient to shoulder-surfing attacks due to the invisible pressure component. To estimate the task overhead introduced by this security feature, we present a comparative evaluation of force-PINs and standard four-digit and six-digit PINs as currently deployed in modern smartphones. We conducted a lab study with $n = 50$ participants to compare four-digit force-PINs against four- and six-digit standard PINs and performed a small shoulder-surfing experiment.

We found that entering force-PINs is more time-consuming than entering digit-only PINs. However, we also found that the difference in authentication time between six-digit and force-PINs was not statistically significant. The number of both critical and standard errors were rather low for force-PINs even though the participants from our lab study were using force-PINs for the first time. According to our survey results, the participants liked the invisible pressure component as an additional security feature.

In a small shoulder-surfing experiment, we found that the force component is more difficult for an attacker to observe: none of the force-PINs entered while being observed by an attacker was guessed correctly. However, the attackers were able to guess some of the digit sequences correctly. We also analyzed the user-chosen force patterns alongside with the entered digits and found that users create higher entropy PINs. In an additional field study, we collected evidence on learning effects and showed that authentication time decreases with training.

In summary, the contributions in this chapter are:

- We propose an enhancement to digit-only PINs with an invisible force component via pressure-sensitive touchscreens.
- We implemented a prototype of the proposed scheme called *force-PINs*.
- We performed an evaluation of force-PINs, including a lab study with 50 participants, a security evaluation, and a field study with 10 participants.

6.2 Related Work

Given the importance and the practical impact, it is not surprising that there has been a significant amount of work on authentication schemes. In the following, we briefly review work closely related to our approach. We also refer to the work by Bonneau et al. [34], who presented a benchmark for evaluating authentication schemes.

Malek et al. [128] proposed a haptic-based graphical password scheme. They complement graphical passwords with personal entropies based on pressure and argue that the password space is increased. However, they did not conduct a user study to evaluate usability factors and do not provide empirical evidence that supports the theoretical calculations of a larger password space. Furthermore, they did not evaluate their approach against a shoulder-surfing threat model.

Bianchi et al. [28, 27, 26, 25] proposed several authentication approaches based on tactile feedback with an emphasis on accessibility and multi-modal feedback. In comparison to our approach, they rely on a tactile wheel to interact with the system, a component which is not available in off-the-shelf devices.

To make smartphone authentication resilient to shoulder surfers, De Luca et al. [51, 53] presented an authentication mechanism that allows users to enter passwords at the front and the back of their device. While their approach offers benefits with respect to shoulder-surfing resilience, a major limitation of this approach is that there is no such device available at this time that provides users a touch-sensitive back.

Harbach et al. [79] performed a real-world study on smartphone unlocking and found that users spend a significant amount of phone usage time on unlocking their device with PINs and unlock patterns. On average, their study participants unlocked their phones about 47 times throughout the day. This finding shows that mobile device unlocking introduces a severe task overhead and highlights that authentication time is an important factor regarding the usability of the method. It also implies that any time-consuming method is potentially disadvantageous for usability and will therefore have difficulties in getting accepted by users. De Luca et al. [50] found that increased authentication time was a reason for Android users to stop using *Face Unlock* (called *Trusted Face* in later Android versions). Their study also revealed that usability factors are the primary

reason keeping users from adopting biometric authentication on mobile devices and that privacy and trust issues only play a secondary role.

A new trending topic in authentication research is implicit authentication. E.g., Buschek et al. [41] studied the feasibility of mobile keystroke biometrics and found that they can be used for user authentication with relatively low error rates. As shown by Khan et al. [106], current methods for implicit authentication are not capable of replacing knowledge-based authentication because their real-world accuracy is significantly lower than in lab settings. Furthermore, they require a certain number of interactions to classify a user correctly. Therefore, these systems are often perceived as disruptive in cases where authentication fails and fallback authentication methods come into play.

6.3 Concept and Objectives

Our approach is based on PIN-based authentication and pressure-sensitive touchscreens as found in modern smartphones (e. g., *3D Touch* available in the iPhone 6s). In the following, we first describe the attacker model and then discuss the design and implementation of *force-PIN*.

6.3.1 Attacker Model

Throughout the rest of this paper, we assume that the attacker is able to perform a shoulder-surfing attack: she is in close vicinity to the user while authentication takes place and can observe the typing behavior (e.g., in a crowded public or semi-public environment). The key element of a successful shoulder-surfing attack is the ability to clearly observe all sensitive information being entered on the touchscreen.

We also assume that an attacker can gain possession of the user's device. In case the device gets lost or stolen, the design of force-PIN makes a PIN harder to guess due to the theoretically larger PIN space and the pressure component.

6.3.2 Force-PIN Design

Force-PINs are designed to be more resistant to observation due to the unobtrusive pressure component that helps to obfuscate PIN components and thereby complements regular PIN entry: a user enters a digit either via a shallow or deep pressure on a pressure-sensitive touchscreen. The user receives tactile feedback when entering a digit with deep force. The tactile component and vibration feedback may implicitly help users to memorize force-PINs [38].

An example force-PIN could be 0-**9**-7-**1** where bold and underlined numbers should be pressed more deeply than others on a pressure-sensitive touchscreen (see also Figure 6.1). The design is not only simple, it is also cheap and easy to deploy as it relies on off-the-shelf hardware. We expect that users who are already using pressure-sensitive touchscreens

will find force-PINs as easy to learn as digit-only PINs as they are based on interactions they are already familiar with.

6.3.3 Implementation

For our study, we implemented a prototype app for iPhones with touch-sensitive screens. The app lets users set a force-PIN and presents a lock screen that looks just like a common lock screen from off-the-shelf iPhones. A force-PIN consists of four digits and a force pattern with two different pressure levels, namely shallow and deep press.

The design decision was based on a small pre-study with 9 participants where we evaluated subjective perceptions on different types of pressure encodings. We evaluated both relative and absolute differences in pressure with different thresholds, respectively. As two-stage pressure with a constant threshold for shallow and deep press performed best; we implemented the prototype app accordingly. We also tested different thresholds and to our surprise it was often not easy to distinguish which threshold was higher and which one was lower. Therefore, we then set the threshold for deep pressure to 50% or more of the maximum possible pressure supported by the hardware.

For our user study, we also implemented apps for four-digit-only and six-digit-only PINs for a comparative lab study and a slightly modified force-PIN app for our field study. The app for the field study had a different main screen and allowed users to submit additional comments to gather in-situ data. Furthermore, the app issued a daily notification to remind the participants of the study task. Each app stored the entered PINs and measured authentication time and failed attempts. The apps with force-PINs also stored the selected four-digit force pattern and arrays of force gradients that were measured for every touch interaction with a pressure-sensitive digit button.

6.4 Lab Study

In the course of a usability lab study, we evaluated force-PINs against digit-only four-digit and six-digit PINs. We chose to evaluate four-digit standard and force-PINs against six-digit standard PINs as they were introduced as the new default in iOS 9. We did not evaluate six-digit force-PINs as we wanted to minimize the additional task overhead. In this section, we describe the methodology and results of this lab study.

6.4.1 Design and Procedure

Our study is based on a within-subjects design, i.e., every participant is exposed to all conditions. This allows us to perform a comparative evaluation of all subjects exposed to our conditions. We assigned every participant a unique ID and a random order of conditions to reduce learning effects.

The three conditions were as follows:

- (C1) four-digit PINs
- (C2) six-digit PINs
- (C3) four-digit force-PINs with shallow and deep pressure

We recruited participants around the university campus over bulletin boards and personal communication mentioning that the study was about their preference of different types of PINs. All of our participants were either employed or currently enrolled as students at the university. We recruited 50 participants for our lab study. They were compensated with a voucher for the university's cafeteria. Table 8.1 shows the demographics of our participants. All participants were frequent smartphone users and had used digit-only PINs before. To reduce the risk of biased interpretation, we presented the three PIN entry methods equally and did not provide any hints on which method was potentially more secure or not. The participants were not told that the study placed an emphasis on evaluating force-PINs.

The lab sessions proceeded as follows: First, the participants were briefed about the purpose of the study. A subsequent training session allowed them to get familiar with the different types of PINs. This was necessary to minimize the bias introduced by the comparison between a well-known and well-trained authentication method and a newly introduced scheme that users have not yet been exposed to.

Then the participants chose a PIN of the first assigned PIN type and afterwards authenticated with the respective PIN until they had completed three successful authentication sessions. After completing this task, the participant proceeded to the next condition, selected a new PIN and authenticated three times. We instructed the participants to select PINs that they thought were as secure as possible and asked them to remember the PINs just like their own ones in real life. We refrained from assigning PINs as it is a common scenario in the smartphone ecosystem that users can choose their own PINs. For the same reason, we did not explicitly disallow PIN-reuse.

The metrics we used for our usability evaluation were authentication speed and error rate as defined by De Luca et al. [51]. They defined *authentication speed* as the time between the first touch and the last touch of the authentication session and only counted successful authentication attempts. Regarding the *error rate*, we differentiate between basic and critical errors (as also proposed by De Luca et al. [51]) where basic errors refer to errors within an overall successful authentication session (failed attempts) and critical errors refer to completely failed authentication sessions. Hence, successful authentication sessions may contain failed attempts that influence authentication speed.

In addition to the data collected through our smartphone apps, we gathered quantitative and qualitative data via a questionnaire consisting of 15 closed and open-ended questions to study the perceived security and usability of the three different types of passcodes. The

reason why we chose to use open-ended questions was that we wanted to collect meaningful participant statements using their own knowledge, perceptions and interpretations. The questions can be found in Appendix 9. After completing the experiments, all participants filled out the questionnaire on a laptop provided by the experimenters.

The participants had to provide their previously assigned experiment ID on the first page of the questionnaire to link the data sets. Except for age, gender and whether the participant had an IT background, no personal data was collected in order to preserve the participants' anonymity. We also collected data on smartphone usage and asked the participants which authentication method they were using at that time on their own smartphones.

The qualitative responses were coded using an iterative coding approach. Two researchers independently went trough the participant responses and produced an initial set of codes. Then, the researchers discussed reoccurring codes, topics and themes, and agreed on a final set of codes. Based on this set, one researcher coded the answer segments for further analysis. As most answers where short and to the point, we did not perform a reliability test of the final coding.

6.4.2 Results

Given our sample consisting of 50 participants, the quantitative results of our study are based on $3 * 3 * 50 = 450$ authentication sessions (three conditions, every pin type was entered three times by 50 participants). Our study has a repeated-measures design, i.e., every participant was exposed to every condition. Therefore, we analyzed our data with repeated measures ANOVAs. We removed 2 authentication sessions that lasted longer than 30 seconds from the dataset as those occurred when participants where distracted from the study task.

Authentication Overhead

Authentication Speed. As proposed by De Luca et al. [51], we measured authentication speed from the first to the last touch of a successful authentication session. Hence, an authentication session can also contain a maximum of two failed attempts. After the third failed attempt, the user was locked out of the app. The participants had to start the sessions by clicking on a button.

We only considered successful authentication sessions to measure authentication speed. As every user entered every PIN type three times, we calculated the average authentication speed for every user and every authentication method and used this value for further analysis. Overall, 56 force-PINs were selected by our participants. Five of them decided to change their PIN during the experiments, one participant renewed the PIN twice. The participants did not mention any reasons for these decisions. The authentication time was measured based on the most recently selected PIN. Table 6.2 shows the mean authentication time in seconds and error rate. Figure 6.2 shows the collected authentication speed measures for all participants and PIN types.

Table 6.1: Participant characteristics from the lab study. n=50

Demographic	Number	Percent
Gender		
Male	31	62%
Female	19	38%
Decline to answer	0	0%
Age		
Min.	19	
Max.	56	
Median	25	
IT Background		
Yes	4	8%
No	46	92%
Smartphone		
Android	32	64%
iPhone	14	28%
Windows Phone	2	4%
Other	2	4%
Used Authentication Method		
4-digit PIN	26	52%
6-digit PIN	2	4%
Password (digits/characters)	3	6%
Unlock Pattern	14	28%
Fingerprint Sensor	7	14%
Face Recognition	0	0%
Android Smartlock	1	2%
None	5	10%

To reveal significant effects regarding authentication speed, we performed a one-way repeated-measures omnibus ANOVA across the 3 PIN types. The results show significant differences in authentication time ($F_{2,147} = 10.19, p < 0.001$). A pairwise t-test with with $t_{0.95,98} = 1.9845$ revealed significant main effects comparing the authentication speed of four-digit with six-digit PINs ($p < 0042$). In addition, authentication speed of four-digit PINs was significantly faster than of force-PINs ($p < 0.001$). The difference in authentication speed between six-digit and force-PINs was not statistically significant ($p = 0.12$).

Errors. An important factor when estimating the overhead of an authentication method is the number of errors. Similar to De Luca et al. [51], we distinguished between basic and

Table 6.2: Mean authentication time in seconds and error rate with different levels of the independent variables.

Authentication Speed	**Mean**	**SD**
4-digit	2.34	1.21
6-digit	3.33	1.56
Force	3.66	1.96
Error Rate	**Basic**	**Critical**
4-digit	21	0
6-digit	22	0
Force	36	4

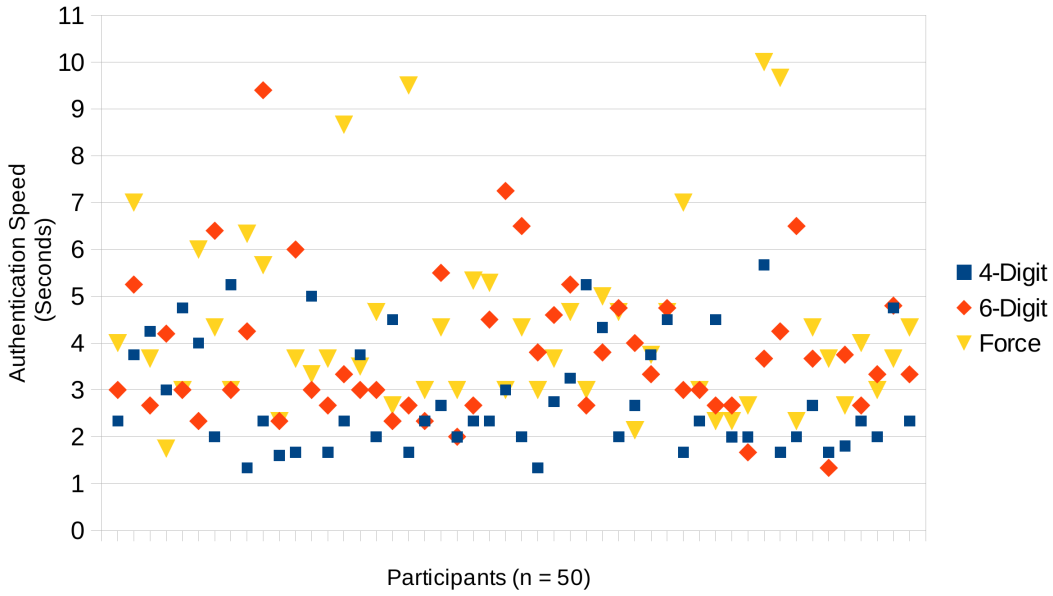


Figure 6.2: Mean authentication time per participant.

critical errors. For our authentication scenario, we defined a *basic* error as an erroneous attempt to enter a PIN code. An authentication session can be successful overall, but may take a user two or three times to enter the PIN correctly. We considered an error as *critical* if the entire authentication failed, i.e., a user was locked out after three erroneous attempts as commonly deployed in off-the-shelf smartphone operating systems.

Out of 450 total authentication sessions, four authentication sessions failed (0.9%). All failed sessions involved force-PINs. 36 (8.0%) failed attempts (basic errors) were registered with force-PINs. 22 (4.8%) failed attempts were registered with six-digit PINs and 21 (4.6%) with four-digit PINs.

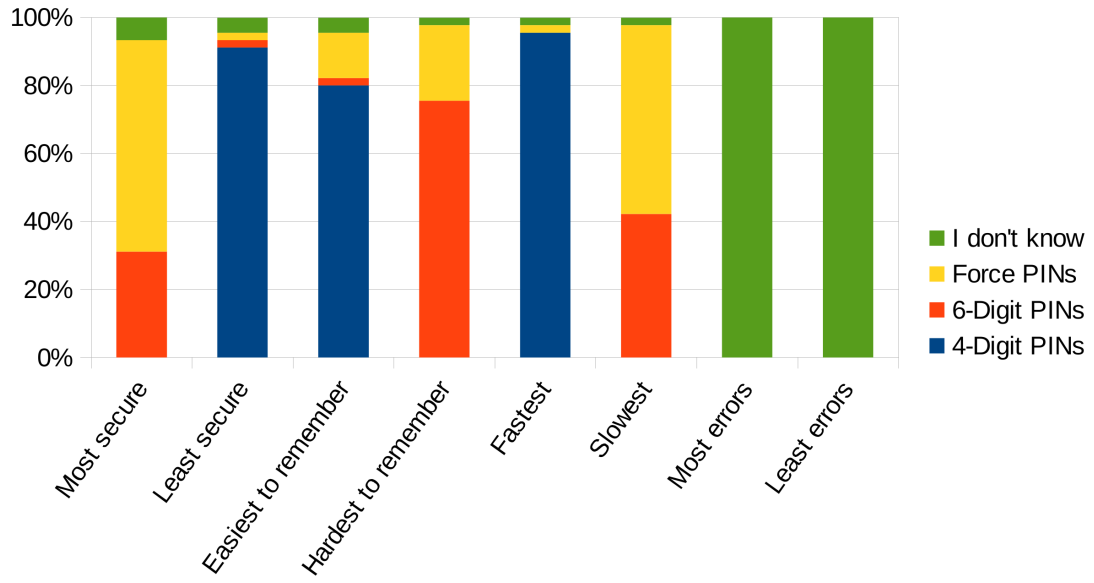


Figure 6.3: Self-reported usability and security estimation in percent.

Perceived Usability and Security

As explained above, participants were asked to fill out a short questionnaire after completing the PIN selection and authentication tasks. In addition to the measurements collected via our iPhone apps, we were interested in participants' perceptions of the three suggested PIN types regarding usability and security. We presented users with closed-ended questions asking which PIN type they thought was the easiest/hardest to remember, fastest/slowest and most/least error-prone to enter and generally most/least secure. The results of these questions are shown in Figure 6.3.

91% of our participants reported that they thought four-digit PINs were the least secure of the three tested PIN types. 95% also thought that four-digit PINs were the fastest PIN type to enter and 80% thought that they were the easiest to remember. 62% thought that force-PINs were the most secure of the three methods but 55% also thought that this was the most time-consuming PIN type to enter. In comparison, only 31% thought that six-digit PINs were the most secure but 75% also thought that they were the hardest to remember.

To our surprise, all participants chose the "I don't know" option regarding most and least errors when entering any of the suggested PIN types.

On the last page of the online survey, we asked participants three open-ended questions related to their perception of force-PINs. This was the only part of our study where force-PINs received particular attention. These questions were asked at the very end of our lab sessions to minimize the risk of biased interpretation.

After coding the data segments collected through these questions, we found that 38 of the 50 participants thought that a major benefit of force-PINs was the resistance against observation due the haptic and invisible component. 10 participants also stated that they think force patterns are easier to remember than additional digits, as would be the case with longer PINs. Eighteen participants reported that they still think that it requires additional effort to enter digits with different levels of force as they are still not used to this new interaction method with touchscreens.

Informal Participant Statements

In this section, we present informal participant statements and also quote some of the qualitative statements gathered via the open-ended questions from our post-experiment survey. These direct quotes are presented as they were given by the participants prior to coding.

Overall, we were surprised by how easy it was to recruit participants irrespective of the promised reward. We had the impression that all of them found the topic of PIN security important. Based on their comments, we had the impression that most of them seemed to be aware of the richness of private data stored on their smartphones. Most participants also asked for further help in protecting their devices after participating in our study. After their participation, they were given the opportunity to have their questions answered by the experimenters. Even though a few authentication sessions with force-PINs failed, all participants understood the concept of force-PINs and were able to use them. To our surprise, the participants found the concept natural and intuitive even though most of them were using pressure-sensitive touchscreens for the first time.

- *"I like the additional dimension. It is invisible and therefore makes my PIN more secure." (P5)*
- *"If someone observes me entering my PIN, which is not that secure and probably easy to guess, at least the force component is harder to guess. (P28)"*
- *"I think it might take a while to fully get used to it, as this concept is new to me. (P23)"*
- *"Why not use a six-digit force-PIN? (P12)"*

Force Pressure

As stated in Section 6.3, we based our design for a two-step scale on our pre-testing with people who had never used *3D Touch* before. Due to the low experience with pressure-sensitive screens, they could not easily distinguish different thresholds to separate deep and shallow press. The app also provided vibration feedback as soon as the user entered a digit with force. Through our lab study, we collected the exact values of the force registered by the device and then used it to evaluate how close or far the registered force

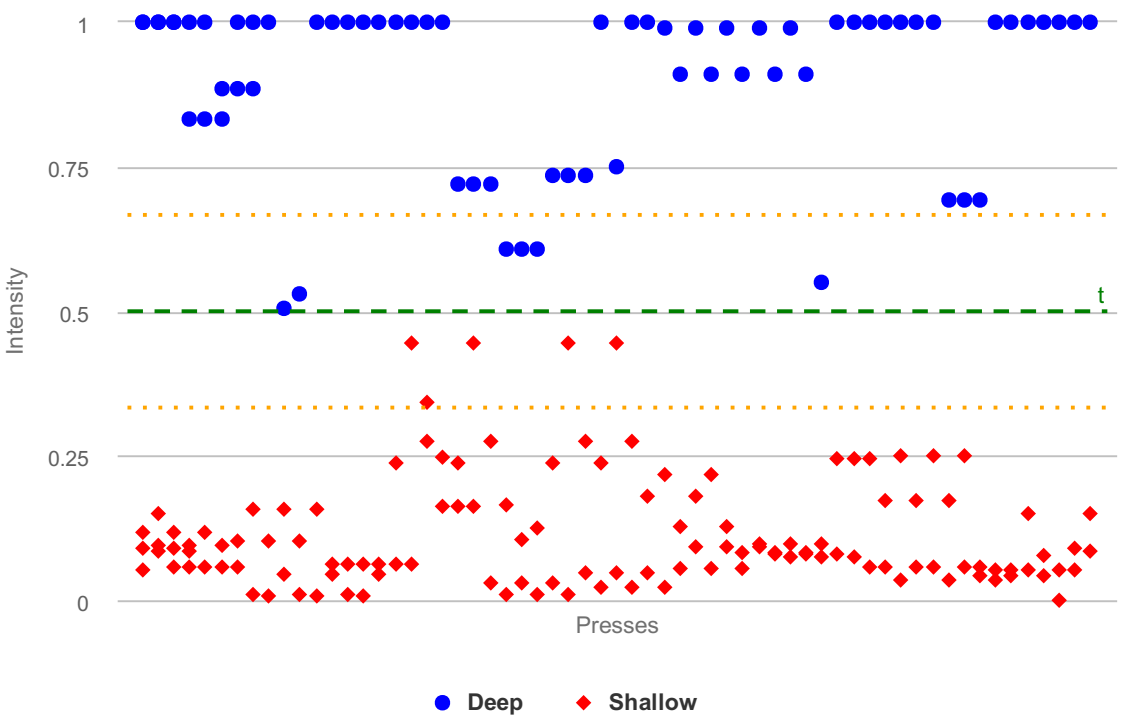


Figure 6.4: Measured force relative to the maximum possible force. The green line at y = 0.5 represents the threshold for distinguishing between deep and shallow presses. The grey lines at 0.25 and 0.75 indicate two potential thresholds for a three-step force scale (e.g., *shallow-medium-deep*.)

was from the threshold and the upper and lower boundaries. Figure 6.4 shows the force intensities of all logged force-PIN digits during the lab study in percent of the maximum possible force.

6.5 Security Evaluation

Based on the data collected during the lab study, we performed an additional security evaluation to evaluate shoulder-surfing resistance and PIN entropy.

6.5.1 Shoulder Surfing

To evaluate our approach to the attacker model, we performed a small shoulder-surfing experiment in the lab. Similar to the study design of De Luca et al. [51] and von Zeschwitz et al. [187], the attacker tried to shoulder surf the force-PIN entry from the victim. For our evaluation, we considered direct observation, i.e., the attacker was physically standing behind the victim and tried to guess the entered force-PIN and then performed an additional evaluation based on separately recorded video material. Our evaluation is

based on the 50 force-PINs which were collected in the course of our lab study and then used for our evaluation of authentication speed and error-rate.

The direct observation attack was performed during the lab study. One experimenter acted as a shoulder surfer and was in close proximity to the victim. Our participants were aware of their entered PINs being tracked via the device used during the experiments but they were not told that one of the experimenters acted as a shoulder surfer. The shoulder-surfing experimenter was perceived as trustworthy. Therefore, the participants did not apply additional measures to prevent their PINs from being observed. We chose this experimental setting as we believe that situations where victims are not aware of being observed are the most dangerous. We furthermore believe that any authentication method should be resilient to direct observation regardless of a specific situation and the user's awareness. In addition, an experimenter entered the collected PINs with their corresponding force patterns while being filmed. Each PIN was entered only once. Another two volunteers, who were university students (one male, one female), then tried to guess the force-PINs based on the recorded material. Each of them tried to guess 25 PINs. They were allowed to re-watch the video sequence up to 5 times if they wanted to.

This first look at shoulder-surfing resistance suggests that force-PINs are capable of making digit PINs more resilient against shoulder-surfing attacks. Out of the 50 entered force-PINs, the shoulder surfer was not able to guess a single one completely. However, 21 out of 50 PINs were partially guessed (i.e., the attacker correctly guessed the digits but not the force pattern). Similar to the direct observation attacks, the attackers in the camera-based attacks were not able to completely guess the force-PINs from the recorded material, but managed to guess 39 of the shown digit sequences correctly. We did not evaluate whether individual digits (with or without force) were guessed correctly.

6.5.2 Entropy

In theory, the PIN space of four-digit force-PINs is larger than for standard four-digit and smaller than six-digit PINs. In our lab study, we used user-assigned PINs. We gave participants a password policy, namely to choose a PIN that, in their opinion, is as secure yet as memorable as possible and where at least one digit within the four digit pattern is entered with a deep press.

Obviously, the number of possible combinations is $10^4 = 10,000$ for four digit passwords and $10^6 = 1,000,000$ for six digit passwords. Force-PINs augment the four-digit password space to $20^4 = 160,000$ possible PIN codes including four-digit PINs with all digits entered with shallow pressure. As we defined a policy for the lab study which forced participants to choose at least one digit with deep pressure, the password space decreases to $150,000$.

As done by Cherapau et al. [43], we calculate the zero-order entropy, which is a theoretical measure of the entire search space of all possible secrets of a given length and the size of a given alphabet assuming that each character is selected randomly. Zero-order entropy is measured in bits and calculated as $L * \log_2 N$, where L is the length of the secret and

N the size of the character set. Hence, for force-PINs, the length is 4 and the character set 20. Thus, the zero-order entropy for force-PINs is 17.28 bits, while four-digit PINs have a zero-order entropy of 13.28 [43] and six-digit PINs 19.93 bits. These theoretical measures are upper bounds for real-world entropy.

In theory, the augmented PIN space is a major improvement compared to standard four-digit PINs. In practice however, users often do not fully exploit this benefit but select PIN codes and passwords from a much smaller subset that are often easy to predict [105]. Therefore, the search space is smaller and the PIN is therefore easier for an attacker to guess. We therefore evaluate the distribution of force patterns and digit-pressure combinations.

Table 6.3 shows the occurrences of force patterns selected by our participants. Our results suggest that more than half of our participants selected a force pattern where only a single digit is entered with deep press. In our sample, the most popular positions in the digit sequence were the first and second one with a probability of 14.0%. Even though this trend indicates that our participants did not fully make use of the theoretically larger PIN space and therefore create lower entropy PINs in practice, this is already an improvement over standard four-digit PINs. Our dataset of 56 PINs is relatively small and therefore not sufficient to determine the practical entropy of force-PINs. To provide a rough indicator, we calculate the entropy of the binary force component based on the force-PINs chosen by our study participants. Furthermore, to estimate the entropy gain over digit-only PINs, we compare our results to those from a related study on iPhone passcodes with a larger sample size. In theory, if force patterns were evenly distributed, the theoretical entropy gain would be 4 bits. We calculate the practical entropy gain as $-\sum_{i=1}^{n} p_i * log_2(p_i)$ where p_i is the probability of a certain pattern occurring. Based on our observed probabilities from 56 user-chosen force patterns (as presented in Table 6.3), the practical entropy gain is 3.41 bits. Bonneau et al. [36] calculated the entropy of four-digit PINs from iPhone users as 11.42 based on a dataset of 204,508 PINs. Comparing our findings with Bonneau et al. [36], an additional binary force component provides an entropy gain of approximately 23% to digit-only PINs of length 4.

6.6 Field Study

In addition to the lab study, we conducted a field study to show that authentication time for four-digit force-PINs decreases with training. The latter is an important metric when comparing the usability performance of digit-only PINs with force-PINs as we assume that users will initially perform better with digit-only PINs as they are already trained to use them.

6.6.1 Study Design and Procedure

We recruited 10 participants and deployed an iOS app on their personal devices and asked them to enter as many force-PINs as possible (we required a minimum of 300

Table 6.3: Force patterns selected by the lab study participants where S = shallow press, D = deep press. n = 56 user-selected PINs. The table is sorted in descending order. The pattern SSSS was excluded as the PIN selection policy required participants to enter at least one digit with deep press.

Force Pattern	Number	Percent
DSSS	8	14.0%
S**D**SS	8	14.0%
SSS**D**	7	12.2%
SS**D**S	6	10.5%
DSS**D**	6	10.5%
S**DD**S	5	8.7%
DDDD	5	8.7%
SS**DD**	4	7.0%
S**DDD**	2	3.5%
S**D**S**D**	2	3.5%
DDSS	1	1.7%
DS**D**S	1	1.7%
DDDD	1	1.7%
DDDS	0	0.0%
DS**DD**	0	0.0%

successful authentication sessions) over a period of two weeks. At the end of this period, we conducted short debriefing interviews with the participants. In contrast to the lab study, the participants were aware that the focus of the study was to evaluate force-PINs.

Due to the low propagation of compatible iPhones in our region, we were able to recruit only 10 participants. In spite of the relatively low number of participants, we still believe that the gathered data provides useful insights and rough indicators on learning effects. Furthermore, deploying force-PINs in a real-world environment helped us to gather in-situ reactions on authentication problems with force-PINs.

We based our study design on findings from Harbach et al. [79], who found that users unlock their phone on average 47.8 times a day (about three unlocks per hour assuming a user is awake for 16 hours per day).

Due to the restrictions in iOS, we were not able to replace the actual PIN scheme on the participants' devices with force-PINs. We also had to reject our plan to issue notifications based on the participants' unlocking behavior as iOS does not offer to activate third-party apps after an unlock event. Therefore, we were not able to collect the respective data from the users' own devices. As everyday routines and smartphone usage habits are highly diverse, we refrained from requiring force PIN entries at fixed time-points throughout the day and opted for a more realistic and less disruptive setting. To evaluate different timing options for notifications, we conducted a small pilot study

Table 6.4: Digits and their occurrence entered with either shallow or deep press. Deep pressed digits are in bold; sorted in descending order.

Digit (shallow/deep press)	**Number**
1 (shallow)	27
0 (shallow)	22
5 (shallow)	16
4 (shallow)	15
3 (shallow)	14
2 (shallow)	12
0 (deep)	12
1 (deep)	12
6 (shallow)	11
2 (deep)	10
9 (deep)	10
3 (deep)	9
6 (deep)	9
9 (shallow)	8
4 (deep)	7
7 (shallow)	6
7 (deep)	6
8 (deep)	6
5 (deep)	6
8 (shallow)	5

with different notification patterns. The participants from this pilot study perceived the notifications as disruptive and annoying regardless of whether they were issued at fixed or adaptive time points. Based on the participants' responses, we decided to reduce the number of daily notifications to a single daily reminder at an arbitrary point in time and left it up to the participants when and how often to entered their force-PINs. We are confident that this study design reflects realistic usage habits and reduced the risk of participants dropping out early from the study.

We instructed our participants to enter force-PINs whenever they took out their phone before or after their primary task. We suggested they distribute the PIN entries over the given period of time (i.e., about 20 PINs a day), but also told them that it was their own decision when exactly and how often to enter them. The participants were also instructed to choose as secure and memorable PINs as possible with at least one digit entered with force.

The main screen of our app had a button that redirected the participants to a lock screen to start an authentication session with a force-PIN. It was designed to look exactly like the standard iPhone lock screen. Our app also displayed a counter of successful authentication sessions and provided users with two extra buttons, one to send us an e-mail in case of questions and another one to leave a comment to a situation. We also provided users with an option on the main screen to set a new force-PIN. Upon clicking on this button, a password-forgotten event was logged and the participants were able to set a new force-PIN.

6.6.2 Results

Table 6.5: Summary of field study results. n=10

Subjects	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
Completed Authentication Sessions	534	336	453	387	407	335	210	386	343	357
Basic Errors	13	41	69	20	4	26	16	17	21	27
Basic Error Rate	2.4 %	12.2 %	15.2 %	5.2 %	0.9 %	7.7 %	7.6 %	4.4 %	6.1 %	7.6 %
Critical Errors	0	3	0	0	0	1	1	1	1	0
Critical Error Rate	0 %	0.8 %	0 %	0 %	0 %	0.2 %	0.4 %	0.2 %	0.2 %	0 %
Forgot Force-Pin	0	2	1	0	0	0	0	0	0	0
Force-Pins	**5225**	**0229**	1234	**5795**	**5968**	**0000**	1**7**03	**0**171	22**0**4	9**9**99
	-	0**229**	741**2**	-	-	-	-		-	-
	-	**1397**	-	-	-	-	-	-	-	-

Overall, our participants successfully completed 3,748 authentication sessions with force-PINs. The results are summarized in Table 6.5. Among the successful sessions, 254 failed attempts (basic errors) were registered and five participants had entirely failed authentication sessions (critical errors). The number of critical errors (i.e., failed authentication sessions) was low. The entirely failed authentication sessions were registered at the very beginning of the study. The error rates in Table 6.5 are given in percent of authentication sessions completed by the user. For the quantitative analysis, we removed authentication sessions that lasted longer than 30 seconds from our sample. As observed in our lab study, authentication sessions longer than 30 seconds usually occurred when the participant was interrupted or distracted from the study task.

The mean authentication speed over all authentication sessions was 2.69 seconds (median=2.26, SD=0.59), which is an improvement over the results from the lab study. The shortest authentication session was only 1.02 seconds long. In comparison, Harbach et al. [77] determined the average authentication speed for digit-only PINs as 1.9 seconds.

All participants attended the debriefing session and participated in the debriefing interviews. One participant did not complete the initially requested 300 successful authentication sessions and had only 210 completed authentication sessions. Although this did not meet our desired goal, we included the data and conducted the debriefing interview with the participant as the number of participants was low.

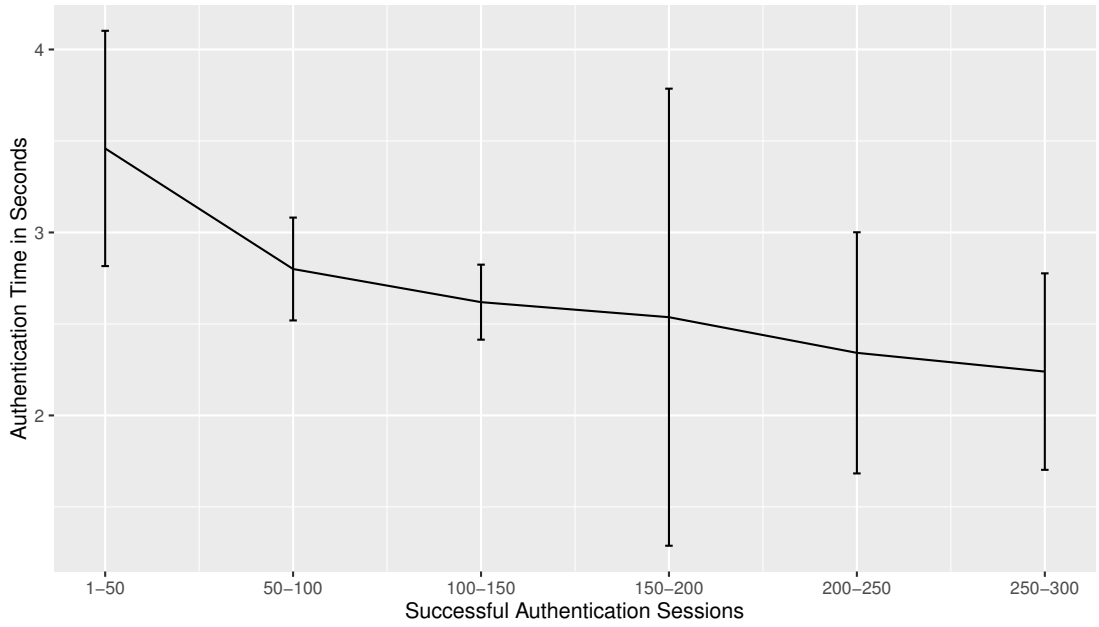


Figure 6.5: Authentication time development based on the first 300 successful authentication sessions across all participants.

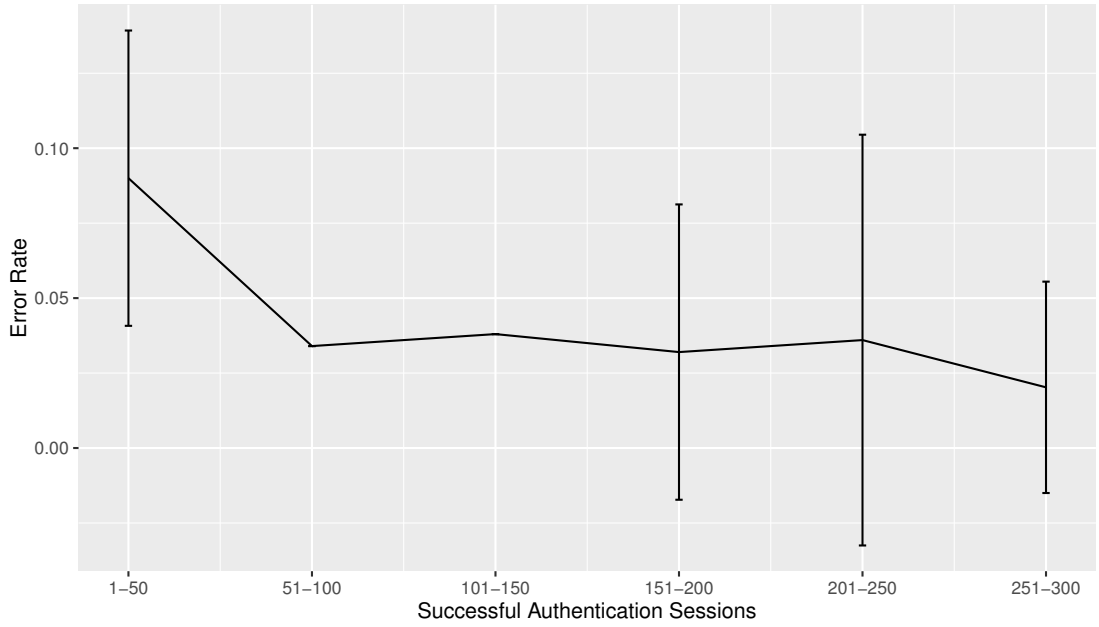


Figure 6.6: Error rate development (basic errors) based on the first 300 successful authentication sessions across all participants.

Just like in the lab study, we measured the authentication time of each session as time from the first touch until the user was successfully authenticated (including potentially unsuccessful attempts made during the session). As per the study design, we expected the PIN entries to be unevenly distributed over time across the participants. Our results show that the participants did not make use of the given time and completed the study task in a few days regardless of our daily notifications. Five participants completed their authentication sessions on a single day. They distributed their PIN entries over the morning, late afternoon and the evening of that day. Four participants completed the study task in two or three days and entered their PINs mostly in the morning and late afternoon/evening of these days. One participant spent four days on the study task and distributed the PIN entries over various times of the day. We therefore refrain from a time-based analysis and compare the results based on authentication sessions.

For our analysis of authentication time and error rate, we consider the first 300 successfully completed authentication sessions from all participants. In order to visualize a trend over multiple completed authentication sessions, we grouped the results in bins of 50 sessions across all participants. We selected a bin size of 50 to approximate the average number of phone unlocks per day as determined by Harbach et al. [79]. We believe that this is a good way to simulate a trend over a reasonable period of time. Figure 6.5 provides a comparison of the average authentication time grouped by 50 successful authentication sessions based on the median authentication time per participant. These results suggest that the authentication time decreases with training. Figure 6.6 shows that the error rate also decreases with training.

6.6.3 Debriefing Interviews

During the debriefing sessions, we asked the participants in which situations they used force-PINs and whether they found them feasible in these scenarios. According to the participants, most force-PINs where entered either while they were at home, in their office, or on public transport. Eight participants reported that they found force-PINs a good way to protect their digit PINs from shoulder surfers even though they estimated their susceptibility towards direct observers as relatively low. Three participants said that they would like to use force-PINs to make their existing PINs more secure against close intruders such as family and friends who could easily guess their PIN as it was an important date. According to them, the risk of a close acquaintance spying on their phones was higher than that of shoulder surfing attacks in public spaces.

Nine participants reported that their perceived authentication time decreased with training when they used it several times a day. However, five of them reported that they still think that simple digit PINs are faster for authentication. All participants reported that they did not find force-PINs harder to remember than simple digit PINs.

Participants were also asked if they would prefer to use force-PINs over simple digit PINs. All of them said that they generally liked the idea of an additional invisible component and six participants said that they would maybe use them if deployed on their device.

Eight participants reported that they found the training phase in the beginning annoying. Three expressed interest in multiple-step pressure difference.

6.7 Discussion

Previous research [79] has shown that the task overhead of smartphone authentication is relatively high. Therefore, we argue that the overhead of a technology to replace simple digit PINs should not be higher than the state of the art.

The results from our lab study suggest that the task overhead of force-PINs is initially higher than for digit-only four- and six-digit PINs. Our security analysis and the participants' responses indicate that force-PINs can increase PIN entropy and improve the resilience towards shoulder-surfing attacks. The results from our field study revealed learning effects after a certain number of interactions with the invisible component, and indicate that authentication time and error rate decrease with training and converge towards the metrics for four-digit PINs.

We collected evidence on frequently used force patterns and determined a practical entropy gain of 3.41 bits based on the force-PINs chosen by our study participants. Similar to other user-chosen secrets, the practical entropy does not meet the theoretical measures but still suggests a major improvement when compared to entropy estimations of digit-only PINs.

Apart from the metrics we used to evaluate the performance of the respective PIN types, the self-reported data from our participants suggests that force-PINs were perceived as more secure than six-digit PINs. The open-ended survey questions revealed that this was mainly due to the force component, which our participants perceived as a good countermeasure against observation.

Only two participants forgot and renewed their force-PINs from the field study. The number of critical errors was also low.

Hence, our results suggest that our scheme is able to improve security with a reasonably low impact on task overhead. In comparison to other solutions, our design improves security without requiring the user to memorize longer sequences of digits, which have been shown to be more difficult to remember [98].

To our surprise, none of the 50 participants provided an estimation of which of the PIN schemes was most/least error prone. While our collected data does not explain reasons, we believe that this is because of the manifold sources of errors: As authentication sessions in the wild usually take place in diverse situations, their successful completion is influenced by environmental and situational constraints beyond the design of the authentication method.

According to a study by Harbach et al. [79], users are generally aware of risky situations but this does not influence their general opinion about this threat, which is that this risk is only considered in a low number of everyday situations. However, just because users

do not perceive situations as risky does not mean that they are not. Hence, physically shielding the PIN from an observer can only mitigate an attack if the user is aware of the threat and therefore actively taking precautions. Our results suggest that force-PINs can help to protect users from shoulder surfers regardless of their risk awareness, while minimizing the additional effort the user has to invest.

Modern smartphones offer biometric authentication as an alternative. While supporters of these methods often argue that they are harder to replicate and therefore not susceptible to shoulder surfing, it is commonly acknowledged by the scientific community that these methods are non-revocable and can easily be broken [42, 16]. Furthermore, they still rely on passwords for fallback authentication. These examples highlight that it is worth putting effort into making knowledge-based authentication resilient to shoulder surfing.

Our prototype app was implemented for iPhone 6s. Other smartphone models, such as the Huawei Mate S, also have pressure-sensitive screens and are therefore suitable for force-PINs. Furthermore, force patterns like in force-PINs can also be added to character/digit passwords with variable length and Android unlock patterns to make them resilient to shoulder surfing attacks. As future work and as soon as a compatible API and device are available in our region, we plan to evaluate force patterns in combination with unlock patterns and other alternative authentication schemes, respectively.

6.7.1 Limitations

We now discuss limitations of our methodology and the conducted studies.

As we recruited our participants at the university campus, the level of education and technology affinity among our sample were higher than expected from the general population. As the results might differ for other demographics, our results cannot be generalized to the entire population of smartphone users. Since only 28% of the participants in the lab study were iPhone users, we cannot determine whether the measurements based on their input were biased by the lack of practice. However, as this study had a repeated-measures design, we were able to perform a comparative evaluation of all subjects exposed to our conditions. All participants in our field study took part with their own devices and had therefore been exposed to a force-sensitive screen before and were already familiar with the iOS user interface and lock screen, respectively.

It is possible that users would improve even more over a longer period of time and usability metrics converge to those of four-digit standard PINs. Regardless of our suggestion to distribute the authentication sessions over the two weeks, the participants tried to complete the study task as fast as possible and therefore entered all force-PINs within the first three days. Also, the number of successful authentication sessions varies widely across the participants. As the participants did not spread out the PIN entries over the given time, we can neither perform a time-based evaluation nor seriously evaluate memorability. The fact that our participants from the lab study thought that force-PINs are more memorable speaks for the system but does not obviate the need for a future long-term evaluation. Regardless of these limitations, we are confident that our study

design reflects real-world usage behavior and due to its flexibility ensured that participants would not drop out early.

A major limitation of this work is that the participants from both the field study and the shoulder surfing experiment participated voluntarily and did not receive a compensation for their participation. Therefore, the motivation for the shoulder surfers was rather low to actually break the system. Another limitation is that they were new to the concept of force-PINs and therefore perceived the task as particularly challenging. Also, force-PINs do not provide visual feedback and the vibration for digits entered with force is very subtle and therefore not audible on the video material. The participants reported finding it hard to focus on both the digits and the force patterns. The person who entered the PINs in front of the camera was a faculty member who was aware of the hypothesis being tested just like the experimenter who tried to shoulder surf the PINs from the lab study. These limitations imply that further investigation is needed. Therefore, we are currently conducting a more thorough study to determine a lower bound for shoulder-surfing resistance.

6.8 Ethical Considerations

Our university does not have an ethics board but has a set of guidelines that we followed in our research. A fundamental requirement of these guidelines is to preserve the participants' privacy and to limit the collection of person-related data as far as possible. For both our studies, we did not collect any personally identifiable information, except for age and gender. A major ethical challenge was the collection of PINs. The PINs were chosen by the participants and they were aware that the PINs they selected were being collected. However, we cannot preclude that those were real PINs. Keeping this data confidential and making it impossible to map a physical person with a certain PIN was therefore our primary concern. In similar shoulder surfing studies, participants were re-corded with video cameras to perform attacks based on the recorded material. Although this was our initially planned study setting, we decided not to film the participants directly while they entered their PINs. This decision was made based on the results and feedback from our pilot study, where our participants expressed discomfort about being filmed while entering information as sensitive as a PIN. We therefore chose to let a separate person enter all force-PINs in front of a camera and then used the resulting material for our camera attacks.

6.9 Conclusion

In this work, we proposed integrating pressure-sensitive touchscreen interactions into knowledge-based authentication. These force-PINs enhance digit-only PINs with a force pattern, i. e., an additional pressure-sensitive component that allows users to select higher entropy PINs that are harder for a shoulder surfer to observe.

We were able to collect evidence on the security benefits of force-PINs and their impact on usability. We conducted a lab study with 50 participants and showed that authentication speed of force-PINs is not significantly slower than that of six-digit standard PINs, but still significantly slower than that of 4-digit standard PINs. We also showed that the error rate is rather low in spite of the fact that most participants had not yet been exposed to pressure-sensitive touchscreen interaction. Furthermore, we conducted a small shoulder-surfing study where an attacker tried to observe and guess force-PINs. The attackers were not able to guess a full force-PIN consisting of a digit sequence and a force component. These results suggest that force-PINs can help to mitigate shoulder-surfing attacks in public spaces that are potentially noisy and crowded. In a security evaluation of the collected force-PINs, we showed that the practical entropy is still higher than for standard four-digit PINs although users do not make full use of the larger PIN space. In an additional field study with 10 participants, we deployed force-PINs in the wild and showed that users improve after being exposed to the technology over a longer period of time.

Our results imply that small enhancements such as an additional pressure component allow users to select higher entropy PINs that are more resilient to shoulder-surfing attacks, while keeping the impact on usability metrics such as authentication speed and error rate low. This is important as users enter their PINs multiple times a day and therefore require methods that do not increase the task overhead.

CHAPTER 7

User Experiences with Bitcoin Security and Privacy

In this chapter, wde present the first large-scale survey to investigate how users experience the Bitcoin ecosystem in terms of security, privacy and anonymity. We surveyed 990 Bitcoin users to determine Bitcoin management strategies and identified how users deploy security measures to protect their keys and bitcoins. We found that about 46% of our participants use web-hosted solutions to manage at least some of their bitcoins, and about half of them use exclusively such solutions. We also found that many users do not use all security capabilities of their selected Bitcoin management tool and have significant misconceptions on how to remain anonymous and protect their privacy in the Bitcoin network. Also, 22% of our participants have already lost money due to security breaches or self-induced errors. To get a deeper understanding, we conducted qualitative interviews to explain some of the observed phenomena.

This chapter is an extended version of [117]. Sections 7.1 provides an introduction. Sections 7.2 and 7.3 provide background information on Bitcoin and related work. Sections 7.5 and 7.6 describe the methodology we used to collect and analyze the data. In Section 7.7 we present the results. Section 7.8 discusses and Section 7.9 concludes our work.

7.1 Introduction

With a current market capitalization of more than 3.5 billion USD, Bitcoin is the most successful cryptographic currency at this time. Bitcoin is utilized for roughly 130.000 transactions per day [33] and has gained significant news coverage. With the success of Bitcoin, several other cryptographic currencies were developed either based on Bitcoin or from scratch.

Although the popularity of cryptographic currencies is increasing, they are not yet a mass phenomenon. One of the reasons is that Bitcoin forces its users to deal with public key cryptography. Furthermore, Bitcoin shifts the responsibilities for most security measures to the end user compared to centralized monetary systems. Even though there is a great variety of software available for managing bitcoins, user-experience is still not obviating the need to deal with the technical fundamentals and to perform backups to recover their virtual monetary assets in case of a loss. Hence, these systems are not resilient to human errors. Reports from online forums and mailing-lists show that many Bitcoin users already lost money due to poor usability of key management and security breaches such as malicious exchanges and wallets. This motivates our research on human interactions with the Bitcoin ecosystem.

Bitcoin users have a huge variety of tools available to manage their virtual assets. These tools are commonly referred to as *wallets*. A wallet was originally defined as a collection of private keys [61]. Hence, a piece of paper with a private key on it or even a mental representation can be considered a wallet. However, most of these tools provide functionality beyond storing keys, such as performing transactions. In contrary to other public key crypto-systems, e.g. PGP/GPG, Bitcoin is not fully communication channel agnostic. In case of Bitcoin the interaction with the Bitcoin network is an integral part to operate in the distributed system. In contrast to other signing systems, Bitcoin tools need to keep state information on performed transactions and account balances respectively.

As a first step to accommodate these misconceptions on Bitcoin wallets, we introduce the term *Coin Management Tool (CMT)* as an extension to the current narrow definition of a wallet. We define a CMT as a tool or a collection of tools which allows users to manage one or more core tasks of cryptocurrencies. Throughout this chapter we are therefore referring to *Bitcoin management*, as it better describes user activities when interacting with the Bitcoin ecosystem. Bitcoin security and privacy aspects have aleady been studied in the research literature [35, 68, 82, 72, 73]. A first look on the usability of Bitcoin key management has been presented in [61]. However, we are the first to conduct a comprehensive user study to collect evidence on user experiences with Bitcoin security and privacy.

In this chapter, we present a comprehensive user study ($n = 990$) to cover human-computer interaction aspects of the Bitcoin ecosystem. The goal was to understand how users interact with Bitcoin and how they manage their virtual assets. We furthermore studied experiences and perceptions related to security, privacy and anonymity in the Bitcoin network. To collect user-reported data, we conducted a comprehensive online survey with 990 participants and qualitative interviews with a subset of 10 participants. Additionally, we extended the evaluation criteria from [61] and provide a method to categorize CMTs depending on the level of control and verifiability a user can exercise with the respective client.

We gathered interesting insights on how users interact with the Bitcoin network and what privacy and security measures they deploy to protect their keys and coins. We found that the first- and third-most used CMTs (Coinbase, Xapo) are web-hosted tools where

users shift security responsibilities to a third party. We also found that about a third of their users are not aware whether their CMT data is encrypted or backed-up. Among the participants who use a web-hosted solution, 50% indicated to use it exclusively while the other half used additional local clients to manage their coins. Regarding risk scenarios and their likelihood to occur, the second-highest risk was attributed to vulnerabilities in web-hosted CMTs (after value fluctuation and followed by theft via malware).

We also found that many users have misconceptions about how to remain anonymous. About 25% of our participants reported to use Bitcoin over Tor which has already shown to be disadvantageous in certain cases [30, 13]. 22.5% of the participants reported to have lost their bitcoins due to security breaches. About half of them consider this loss as their own fault and the majority of them was not able to recover their bitcoins and lost money permanently. Our work contributes research on user-centric concerns of Bitcoin management, as according to Bonneau [35] Bitcoin is one of the cases where practice is ahead of theory.

7.2 Background

The Bitcoin currency is based on a distributed P2P system which synchronizes a public ledger of all transactions among all Bitcoin clients. As a consequence, every full client in the Bitcoin network is able to see the entire history containing all prior transactions. Thereby it is possible to determine the current balance of every account. The account information in Bitcoin basically consists of a hash over a public key which can be compared to an account number, the so-called *Bitcoin address*. The protocol does not require a link between account information and personal data. An individual can have more than one account, hence Bitcoin provides a certain degree of pseudonymity [13, 63].

To transfer n bitcoins from account A, which is under control of Alice, to another account B, which is under control of Bob, a new transaction is created by Alice. Thereby, Alice creates a transaction message with the amount of bitcoins she wants to send to Bob and includes the hash of the public key of Bobs account B as a destination before signing it with her secret key sk_A. Alice publishes this transaction in the Bitcoin network so that every participant knows that Alice now has n bitcoins less on her account A and Bob has received the difference on his account B. When this transaction is successfully propagated in the network, Bob can create new transactions from his account B to another account and spend the previously received bitcoins. This chaining mechanism works fine for passing over arbitrary amounts of bitcoins from one account to another, except in the special case of the first transaction in a chain, because this is where new bitcoins come into existence [141].

Bitcoins are created during the so-called *mining process*. In this procedure every miner collects transactions which have recently been propagated in the P2P network. Then they try to successfully create a new block out of all unconfirmed transactions that have not yet been included in a block of the block chain. A block essentially consists of a

collection of valid transactions[1], a nonce value, and a proof of work. The proof of work is a partial pre-image attack on SHA-256 over the whole block as input. For the attack to succeed, the hash has to be a value smaller than the current difficulty in the Bitcoin network. In other words, the SHA-256 hash has to start with a certain number of zero bits. The number of zero bits is referred to as difficulty. Since SHA-256 is categorized as a cryptographic hash function [143], it is easy to verify a previously calculated SHA-256 sum of a block, but it is considered infeasible to generate a specific block that produces a given hash value. To achieve this, the nonce field is constantly incremented to search for a hash value that fulfils the described property. This brute-force process of searching is called *mining*. If one client in the Bitcoin network finds such a combination of valid transactions and nonce that yields a desired result, he/she publishes this new block in the Bitcoin network and gets rewarded with newly created bitcoins.

The reward comes in form of a new transaction of (currently) 25 bitcoins that has no predecessor and is included as a special so-called *coinbase transaction* by the creator of the respective block. This coinbase transaction also includes the public key/bitcoin address of the creator and marks the first transaction of a new chain of Bitcoin transactions [141, 31, 32, 120].

7.3 Related Work

We build upon already existing work by contributing the first user study with Bitcoin users. Eskandari et al. [61] presented a first look at the key management of Bitcoin by providing a set of evaluation criteria for Bitcoin wallets and a cognitive walkthrough [189] of selected wallets. The work by Eskandari et al. [61] can be considered a first look at the usability of Bitcoin.

Moore et al. [139] conducted an empirical analysis of Bitcoin exchange risks. They examined the track record of 40 Bitcoin exchanges and found that 18 had been closed, with customer account balances often wiped out. They also found that popularity is a strong indicator to predict the lifetime of an exchange, i.e. popular exchanges have a longer lifespan.

Baur et al. [24] conducted exploratory interviews with individuals of distinct groups and found that most stakeholders perceived the ease of use still as rather low. They also found that the experienced usefulness varies according to the user group.

However, no empirical study has been performed to examine user perceptions of Bitcoin security, privacy and anonymity. For a cryptographic currency like Bitcoin, public key cryptography is required. Regarding the usability of key management and encryption in the context of e-mail various studies have shown that there are numerous usability issues regarding the successful usage of public key cryptography [190, 70, 69, 169]. At this time, for neither domain a fully usable concept has been successful. Human aspects

[1] More precisely a Merkle-Tree Hash over those transactions, for details see the specifications [31, 32, 120]

of key management have already been studied in other domains [190, 69, 70, 169, 71]. For the Bitcoin ecosystem however, secure key management alone is not sufficient, as communication is not channel-independent but an integral part of the security concept.

7.4 User Study Methodology

The goal of this study is to empirically investigate end user perceptions and behavior in the Bitcoin ecosystem with an emphasis on security practices as well as coin and key management with the involved security risks. We designed an online questionnaire and additionally conducted qualitative interviews. We derived specific research questions from already existing literature on Bitcoin (as discussed in Section 8.2) as well as from a qualitative content analysis of threads from online forums and mailing lists. Furthermore, we revised the available Bitcoin wallets[2] and their capabilities and used them as inspiration for our questions and the design of the security and privacy risk scenarios. We focus on Bitcoin as it was by far the most popular cryptographic currency at the time we conducted this study (July 2015). While the online survey was intended to broadly measure self-reported Bitcoin management behavior and risk perception, the interviews were conducted to get a deeper understanding on key usability issues, causes of common security incidents and if and how they managed to recover their keys.

7.4.1 Research Questions

We sought answers to the following questions regarding users' perceptions of Bitcoin management and Bitcoin-associated security risks:

- *Q1: What are the main usage scenarios of Bitcoin?*
- *Q2: How do participants manage their Bitcoins? What are participants' current practices and how do they deal with security, privacy and anonymity?*
- *Q3: How do participants perceive Bitcoin-associated security risks?*
- *Q4: What security breaches have affected users and how did they recover their Bitcoin keys and bitcoins?*
- *Q5: What are the main usability challenges that users have to deal with when using Bitcoin?*

7.5 Online Survey

We conducted our online survey over July 8-15, 2015. Our survey consisted of both closed- and open-ended questions and covered the following topics: (1) Bitcoin usage and

[2] `bitcoin.org`

management, (2) CMT choice and usage, (3) security, privacy, anonymity and backup behavior, (4) risk perception, and (5) demographics. The full set of questions is presented in Appendix 9. The open-ended questions were coded independently by two researchers independently. After agreeing on a final set of codes, we coded all answer segments for the final analysis. Coding refers to categorizing qualitative data to facilitate analysis [123] and is a common practice in human-computer interaction research.

7.5.1 Recruitment

We hosted our survey at *soscisurvey.de*. To restrict our participants to Bitcoin users only, we deliberately designed our study to exclude all non-Bitcoin users. As it is difficult to construct such a restricted sample on platforms like Amazon Mechanical Turk, we decided to use Bitcoin mailing lists and forums for recruiting. Furthermore, we compensated participants in Bitcoin. The reward for a completed questionnaire was 4.2 mɃ(= 0.0042 Ƀ ≈ 1.22 USD at that time) After completing the survey, the participants were instructed to enter a valid Bitcoin address to receive the payment. This ensured that everyone who wanted to receive bitcoins as a reward is a Bitcoin user and hence exactly our target audience. Even participants who had not used Bitcoin before had to create a Bitcoin address to receive the compensation.

To motivate participants to spread the word and thus recruit further participants, we displayed a link for re-distribution at the end of the survey. All participants that recruited others received an additional 1 mɃ(≈ 0.29 USD). Table 7.1 shows that this additional incentive scheme was successful since we received a high number of participants this way. As Table 7.1 shows, the top 5 re-distributors of the link recruited about one quarter of the overall sample. Initially we distributed the link to our survey over the following channels: *bitcointalk.org* forum[3], *bitcoin-list* mailing list[4], *twitter.com*[5] and an Austrian bitcoin mailing list[6]. We aimed for maximum transparency to avoid that our call for participation would be misinterpreted as scam. Therefore, we proved on the initial page of our survey that we indeed hold a respectable amount of bitcoins[7], by providing our Bitcoin address[8] together with a signature with the according private key (see appendix 9 for the signature).

We recruited 1,265 participants over July 8-15, 2015 via these channels. The total sample size after filtering out 275 participants due to incomplete or duplicated submission, or invalid entries, was 990. Of these, 85.2% claimed to be male (m), 10.5% claimed to be female (f). 4.3% of our participants preferred not to provide their gender. Ages ranged from 15 to 72 (median = 28.56). About half of our participants reported to have an IT-related background. According to the collected IP addresses, most of our participants

[3] https://bitcointalk.org/index.php?topic=1114149.0
[4] http://sourceforge.net/p/bitcoin/mailman/bitcoin-list/?viewmonth=201507
[5] https://twitter.com/bit_use
[6] http://bitcoin-austria.at/
[7] We purchased our 6.3965 BTC at https://coinfinity.co/
[8] https://blockchain.info/address/12yeU5ymM67SL5UWVSwErAgwVwwaTd1Nma

Table 7.1: Most refereed links.

Reference	Occurrences	Reward in BTC / EUR / USD
455975	91	0.0952 / 24.78 / 27.18
1295	58	0.0622 / 16.19 / 17.76
699324	51	0.0552 / 14.37 / 15.76
932181	28	0.0322 / 8.38 / 9.19
637623	21	0.0252 / 6.56 / 7.19

filled out the survey in the US, followed by the UK and Germany. 7.6% accessed the survey site over Tor (Figure 7.1). These numbers can of course be biased by VPN usage.

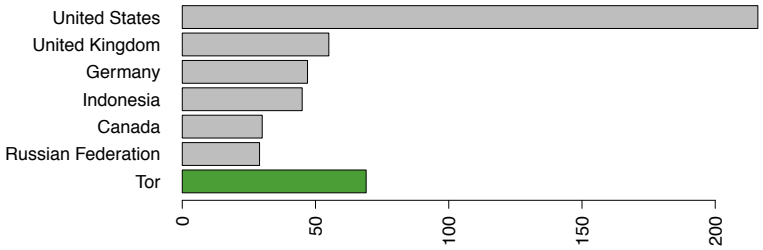


Figure 7.1: Countries from which our participants accessed the survey site.

7.5.2 Validity of our Dataset

Since the survey was designed to be anonymous and we only required a valid Bitcoin address, we had to take special care to avoid abuse. We semi-automatically verified the authenticity of our dataset and were able to exclude 116 submissions we suspected to be fraudulent, and 160 incomplete submissions. Nevertheless, there is still a chance that we missed some manual double submissions. However, due to our deployed countermeasures and the high quality of submitted data (e.g., the open-text questions) we suspect that the overall number is negligible. Among other, we deployed the following countermeasure to make automation harder: ***reCAPTCHA:*** The last page of our survey contained a text box to enter a Bitcoin address for receiving the compensation and a Google reCAPTCHA. This together with the relatively low overall amount of compensations helped to mitigate fully automated submissions. Since reCAPTCHA adapts the difficulty depending on the source IP address, some Tor users complained about hard-to-solve CAPTCHAs. ***Meta data:*** The meta data like source IP address and information on the user's browser was used to pinpoint simple double submission attempts. ***Time:*** We considered submissions below a certain threshold fraudulent since it is impossible to provide reasonable answers under a certain lower bound. ***Open-text questions:*** In suspicious and borderline cases we manually checked the open-text questions to see if the user had meaningful contributions to the survey. ***Reference links:*** The reference links also provided a good insight when users attempted to submit multiple surveys and always referenced their initial survey. ***Bitcoin address:*** The uniqueness of a Bitcoin address was also an indicator for double submissions.

In case we detected double submissions, we accepted only the first submission for our dataset as well as for our compensation scheme. All subsequent submissions were excluded. In conclusion, we did not encounter fully automated submissions and that most fraudulent attempts can be attributed to simple manual double submissions. Moreover, the Bitcoin community has proven to be very forthcoming. There have been cases in which participants deliberately did not include Bitcoin addresses, and commenting that they would like to help by saving the reward in order to recruit more participants.

The demographics of our sample correspond with data on the general Bitcoin population[9].

7.6 Qualitative Interviews

To get a deeper understanding of the findings from our online survey, we conducted an additional field session with qualitative interviews.

7.6.1 Design and Recruitment

We recruited participants via a local Bitcoin mailing list and conducted a two-hour field session at a local bar that accepts bitcoins. All interviewees are regularly using Bitcoin and had previously completed our online questionnaire. Two researchers were present during the field session, one conducted the interview and the other one took notes. As all participants were very particular about preserving their privacy, we chose not to audio-record the interviews.

For the evaluation of our qualitative data, we focused on the exploration of ideas and insights of the participants. Some of the numbers gathered from the interviews will be used as rough indicators to discuss and complement the results from our quantitative survey. We interviewed 10 participants in total. All participants were male and frequent users of Bitcoin and other crypto-currencies. All of them reported to have an IT-related background. The purpose of the qualitative interviews was mainly to complement our quantitative results and to explain phenomena and trends from our online survey. After 10 participants, we reached saturation and little to no further insights were gained, so we concluded the study.

7.6.2 Coding

After the interviews, we went through the collected data and produced an initial set of codes. We traversed the data segments collected from each participant for each question and also included statements that did not directly evolve from a question. Two researchers performed the initial coding independently of each other to minimize the susceptibility of biased interpretation. After the initial coding process, we revised the retrieved codes and discussed recurring themes, patterns and interconnections. After agreeing on a final

[9] http://www.coindesk.com/new-coindesk-report-reveals-who-really-uses-bitcoin/

set of codes, we coded the entire interview data. We coded all data segments, regardless if they emerged directly from a question or a continuative discussion.

7.7 Results

In this section, we present an analysis of the participants' responses addressing our research questions defined in Section 7.4.1. At the beginning of each section we analyse the results from our online survey, whereas at the end we compare these results with our qualitative interviews and try to correlate and explain our findings.

7.7.1 General Bitcoin Usage (Q1)

Most participants reported to use Bitcoins for tips and donations (38.0%), followed by virtual goods, such as web hosting, online newspapers (33.3%), online shopping (27.5%), altcoins (26.5%), gambling (26.5%) and Bitcoin gift cards (19.9%). About 5% self-reported to buy or have bought drugs with bitcoins. 30.2% of our sample reported to use Bitcoin at least once a week, 25% stated that they use Bitcoin at least once a month and 19% at least once a day. The remainder of the participants indicated to use Bitcoin at least once a year or even less. These results suggest that the majority within our survey frequently uses Bitcoin.

We also asked our participants about the amount of bitcoins they are currently holding. About half of the participants did not want to specify. According to their reports, our sample holds approximately 8000 ฿ in total. The majority of users (70%) started to use Bitcoin between 2013 and 2015. 17% started between 2011 and 2012. 58.0% reported to use other crypto currencies in addition to Bitcoin, most frequently Dogecoin and Litecoin. The most popular Bitcoin exchanges in our sample are BTCE (20.9%), Bittrex (14.0%) and Bitstamp (13.0%). 11.4% of our participants are currently mining bitcoins. Most of them started mining after 2014. Many of those who started earlier have stopped mining as they currently consider it infeasible. 195 (19.7%) participants claimed to be running a full Bitcoin server that is reachable from the Internet. The top-mentioned reason for running a Bitcoin server was to support the Bitcoin network (60.5%), followed by fast transaction propagation (46.6%), network analysis (30.3%) and double-spending detection (26.1%).

All participants from our qualitative interviews are frequent Bitcoin users, and some of them are active in the local Bitcoin association. Most interviewees mentioned that the decentralized nature of Bitcoin was among the main reasons to start using Bitcoin. The second-most mentioned reason was simply curiosity. One participant who used to live in Crimea at the time the Ukrainian-Russian conflict started mentioned socio-political reasons. He used to work for a US company at that time and needed a safe and cheap option to receive his salary in Crimea. He furthermore wanted to make sure to not lose any money due to the annexation to the Russian Federation. In his opinion, Bitcoin was the best option and according to him, many people started using Bitcoin at that

time in Crimea. Some participants also mined Bitcoins some years ago when it was still profitable to mine at small scale.

7.7.2 Practices of Bitcoin Management (Q2)

Bitcoin Wallets and Backup Behavior.

Table 7.3 shows the most widely used Bitcoin wallets. The participants could mention multiple wallets as it is a common scenario that users use more than one wallet. The table also shows the number of participants from our sample who use a certain wallet as well as the percentage. Furthermore, Table 7.4 shows whether the users protect their wallets with a password and if these wallets are encrypted. Our findings show that the majority of users protect their wallets with a password. In case of web clients, we observed a lack of background knowledge. For example, 47.7% of Coinbase users in our sample say that their wallet is encrypted and 34% claim that they do not know if it is encrypted. We observed a similar trend for Xapo which is the third-most used wallet in our sample. Just like Coinbase, it is also a web-hosted tool and, similarly to Coinbase, only about half of the users say it is encrypted and about a third does not know if it is encrypted. Regarding backups, only a third of Coinbase users and 43% of Xapo users backup their wallets. 33.9% of Coinbase and 28.5% of Xapo users do not know whether their wallet is backed up. We also found that Bitcoin users with more than 0.42฿(100 USD) do not backup their CMT more often than users with less bitcoins. This effect is statistically significant in our sample ($\chi^2(1) = 5.1, p = 0.02$).

We also asked our participants whether they create additional backups in case their primary backup gets lost or stolen. In our sample, Bitcoin Core users have the highest rate of additional backups. 64% of them indicated to make a secondary backup of their wallet. Table 7.2 shows self-reported properties of wallet backups. According to our data, none of our participants stores a backup on an air-gapped computer. The most reported backup properties were encryption and password protection. According to our sample, 197 backups are stored in a cloud.

59.7% of our participants only use one wallet to manage their bitcoins. 22.7% use two, and 10.6% use three wallets. The remaining 7% use four or more wallets. The maximum number of wallets a participant reported to use was 14. This participant justified this high number by reporting that he wanted to try out the wallets before choosing those that met his requirements best. About half of our participants who used a web client did this exclusively to manage their bitcoins. The other half used a web client in addition to a local client. To our surprise our results show that most coins of our participants are stored in Armory[10]. The Armory users in our sample have about 3818 ฿ in their Armorys, where the top five users reported to have 2,000 ฿, 885 ฿, 300 ฿, 230 ฿ and 150 ฿. The highest reported number of bitcoins stored in a participant's web client was 100 ฿. The reported sum of all coins stored in Coinbase is 238 ฿, in Xapo it is 157 ฿.

[10] https://bitcoinarmory.com/

Figure 7.2 illustrates the accumulated bitcoins per wallet as reported by our participants.

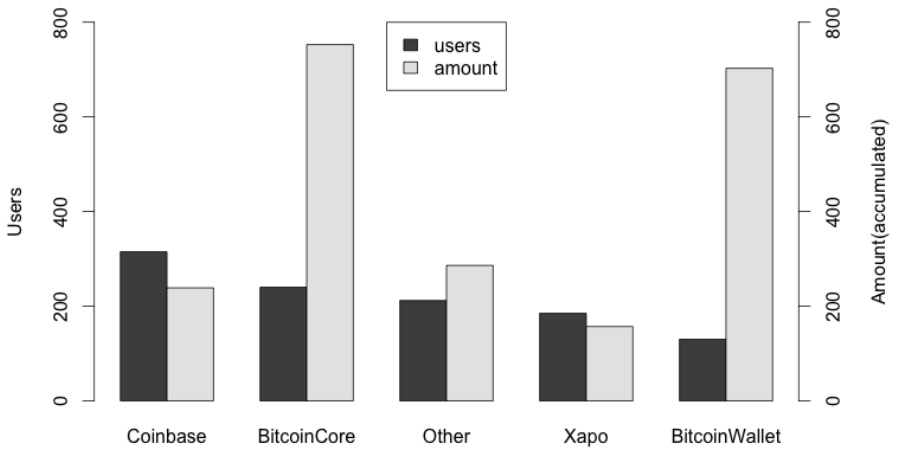


Figure 7.2: Self-reported wallet usage and accumulated hosted bitcoins per wallet.

Table 7.2: Backup properties in absolute mentions in descending order; a user can have multiple wallets and multiple backups.

Backup properties	**Mentions**
My backup is encrypted.	662
My backup is password protected.	629
My backup is stored on external storage (e.g. USB drive).	430
My backup is stored on paper.	334
My backup is stored in the cloud (e.g. Dropbox).	197
My backup is stored on an air-gapped device.	0

Table 7.3: Properties of the most frequently used wallets mentioned by our participants.

CMT	**Number**	**Percent**	**฿**
Coinbase	314	31.7	238
Bitcoin Core	236	23.8	752
Xapo	179	18.1	157
Electrum	125	12.6	226
MyCelium	97	9.8	62

Table 7.4: Properties of the most mentioned CMTs. The three blocked columns contain information on whether the CMT is encrypted, if it is backed up, whether there exists an additional backup and the mentions in percent (Yes, No and I don't know (IDK)). The rightmost column contains the sum of bitcoins stored in a respective CMT by our participants.

	Encrypted?			Backup?			Additional Backup?		
CMT	**Yes**	**No**	**IDK**	**Yes**	**No**	**IDK**	**Yes**	**No**	**IDK**
Coinbase	47.5	18.5	34.0	35.5	30.6	33.9	30.3	66.9	2.8
Bitcoin Core	72.8	16.1	11.1	76.3	14.0	9.7	64.0	32.2	3.8
Xapo	51.4	19.0	29.9	43.0	28.5	28.5	41.3	57.5	1.2
Electrum	72.8	15.2	22.0	77.6	16.0	6.4	55.2	44.0	0.8
MyCelium	61.9	21.6	16.5	83.5	12.4	4.1	52.6	47.2	0.2

Anonymity.

We found that 32.3% of our participants think that Bitcoin is per-se anonymous while it is in fact only pseudonymous. 47% thinks that Bitcoin is not per-se anonymous but can be used anonymously. However, about 80% think that it is possible to follow their transactions. 25% reported to have used Bitcoin over Tor to preserve their anonymity.

We also asked participants if they take any additional steps to stay anonymous. 18% reported to frequently apply methods to stay anonymous on the Bitcoin network. Most of them reported to use Bitcoin over Tor followed by multiple addresses, mixing services, multiple wallets and VPN services. As shown by Biryukov et al. [30, 13] using Bitcoin over Tor creates an attack vector for deterministic and stealthy man-in-the-middle attacks and fingerprinting.

7.7.3 Risk Perception (Q3)

We were also interested in user perceptions of risks associated with Bitcoin. We provided the participants with 11 risk scenarios. We selected the risk scenarios based on findings from scientific literature and evidence from online resources. For each risk scenario, we provided an easy-to-understand description and asked the participants whether they think the risk is likely or unlikely to occur. Figure 7.3 shows the participants' risk estimation. Our results show that the participants consider value fluctuation as the highest risk, followed by vulnerabilities in hosted wallets and Bitcoin theft via malware. Our participants estimated the risk for cryptographic flaws as the lowest, followed by double-spending attacks and DoS attacks on the Bitcoin network.

7.7.4 Security Breaches (Q4)

About 22.5% indicated to have lost bitcoins or Bitcoin keys at least once. Of those, 43.2% mentioned that it was their own fault (e.g., formatted hard drive or lost a physical device with Bitcoin keys). 26.5% reported that their loss stemmed from a hardware failure (e.g.,

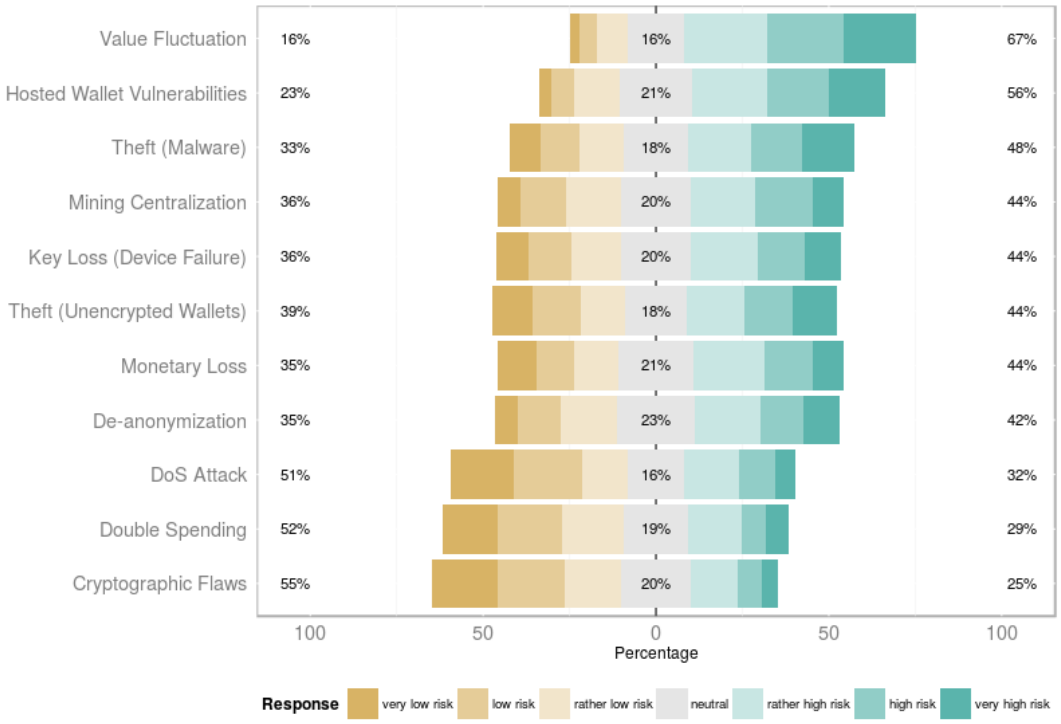


Figure 7.3: User perceptions of risk scenarios in percentage of participants ($N = 990$).

a broken hard drive), followed by software failure (24.4%; e.g. keyfile corruption) and security breaches (18% e.g., malware, hacker).

The majority (77.6%) among those who lost bitcoins did not want to indicate whether they were able to recover their keys. Of those who provided an answer, 65% were not able to recover their keys. Overall, our participants reported to have lost about 660.6873 bitcoins. However, it must be taken into account that we did not ask when the coins were lost. Hence, interpreting this result we must take into consideration that the Bitcoin exchange rate is highly volatile and it is therefore hard to provide an overall estimation in USD. About 40% of our participants reported to have lost money due to a self-classified major security breach. 13.1% of our overall sample reported to have lost bitcoins in HYIPS (high-yield investment programs) and pyramid schemes. 7.9% lost money at Mt. Gox.

We also gave our participants the opportunity to describe how they dealt with the incident. Most participants stated that they did not do anything to recover their keys and simply accepted the loss. Some argued that the financial loss was not worth the effort to take further steps or that they felt helpless as they didn't know what to do. Those who actually took action most frequently mentioned that they filed claims and

contacted the exchange or online wallet provider. Those who lost money to a malicious online wallet reported to have moved to other types of wallets instead of hosted/online wallets. The participants who lost money in HYIPS mostly stated that they started to use less risky investments and learned from their previous mistakes. Irrespective of the security breach, many participants reported to have spread the word over forums on the Internet and shared their experiences with other affected users.

Participant Statements

- *"I follow the 'do not invest more than you're ready to lose' rule."* (P3848)
- *"I just had to accept that my money was stolen ... and that I learned my lesson to never use exchanges as wallets. Keep everything in your own hand."* (P3763)
- *"Just learned from it. It was exceedingly stupid on my part."* (P853)

Eight participants from our qualitative interviews reported that they have already experienced an intentional or accidential key and/or Bitcoin loss. Three participants were affected from the Mt. Gox security breach and two of them reported to have filed a claim on Kraken[11]. One participant reported to have lost a *physical* Casascius[12] Bitcoin but then stopped searching for it as it was only worth about 9 USD at that time. Others also mentioned to have lost their keys due to device failure, corrupted HDDs, or software failure.

7.7.5 Perceptions of Usability (Q5)

Even though most participants of our qualitative interviews were very much concerned about security and privacy aspects of Bitcoin management, eight of them said that they would recommend web wallets and deterministic wallets to non-tech-savvy Bitcoin users. Convenience and easiness of use were highlighted as the main benefits. One participant said that he would definitely recommend a wallet where the private key is stored on a central server to make key recovery easier and to obviate the need for comprehensive backups as well as that mnemonics would help. Six participants also said that they would recommend MyCelium[13] as the most usable wallet. Those who had already used MyCelium consider the paper backup procedure as the most usable and secure way. To create a paper backup with MyCelium, the user has to print out a template that contains some parts of the key and then lets the user fill out the empty spots manually. Some participants expressed initial discomfort when they used paper wallets.

Most interviewees also highlighted the need for fundamental education in early years of childhood. P2 said that Bitcoin is inherently complex, that the fundamental idea of

[11] https://www.kraken.com/
[12] https://www.casascius.com/
[13] https://mycelium.com/

public key cryptography should be taught in school and monetary systems are a matter of culture.

Two participants also highlighted that user interfaces should be simplified and minimalized. Many participants stated that for a fast proliferation of Bitcoin, simple and intuitive UIs are more important than security. They argued that computers proliferated even though most people do not know how computers work and that security is not necessarily an argument for large-scale adoption. They provided examples such as cars in the 1940s, computers, credit cards and WhatsApp. They also said that the amount of money that is circulating in the Bitcoin network is low enough to take the risk of loosing it and compared this scenario to the risk of loosing cash. Some participants also proposed a dedicated device with an intuitive UI for key management and think that such an artifact would be the most secure and usable option.

Participant Statements

- *"It somehow didn't feel right for me to go out of the digital realm."* (P6 on paper wallets)
- *"Children learn about our monetary system in their very early days in primary school. This is why society knows how to use cash and credit cards. I'm sure it could be the same thing with a decentralized crypto-currency."* (P7)

7.8 Discussion

The goal of this work was to answer the research questions provided in Section 7.4.1 in order to understand how users interact with the Bitcoin ecosystem. As this is the first-ever user study focused on user experiences with Bitcoin security and privacy, we gathered useful insights. In the following we discuss our results in the context of already existing works in the field.

Regarding Bitcoin management tools and practices (Q2), we found that two of the most widely used CMTs were web-hosted solutions that obviate the need for users to deal with key management and backups. Our results show that our participants had clear preferences regarding their choice of CMT. In contrary, this is not the case for Bitcoin exchanges. Our data shows that the Bitcoin exchanges chosen by our participants were almost evenly distributed. Even though our data reveals a clear tendency towards web-hosted solutions, these CMTs do not host the majority of our participants' bitcoins. According to our participants' self-reported data, the highest amount of accumulated bitcoins is hosted in Armory. At the time of writing, if used correctly, Armory is one of the most secure solutions.

For the two most widely used web-hosted CMTs, about a third of our participants are unaware of whether their wallet is encrypted or backed up. In such a scenario, users shift

responsibilities to a third party. Even though this seems to be a convenient and usable solution for non-expert users, it implies that the user trusts these third parties to take care of their security. About 50% of web client users indicated to use an additional local client to store their virtual assets. According to our results, users that have a higher number of bitcoins do not necessarily back up their wallets more often. Also, MyCelium users back up their wallets more often than others. Hence we conclude that backup motivation and respectively fatigue depend highly on usability and not on the number of coins.

As the answer to Q4 indicates, participants have already lost money to malicious hosted-wallet providers. Also, our participants perceived vulnerabilities in hosted wallets as the second highest among our risk scenarios (Q5). Some participants from our qualitative interviews said that they would recommend inexperienced users to start with a hosted wallet due to the usability benefits as for most other solutions users are required to have at least a basic understanding of the underlying basics of Bitcoin and the blockchain.

Bitcoin is a pseudonymous system, whereas a wide-spread myth says that it is per-se anonymous. More than a third of our participants still believe in this myth and reported that they think that Bitcoin is fully anonymous. About half of our participants are aware that Bitcoin is not per-se anonymous, but that it can be used anonymously. Regarding anonymity measures, many users reported to use Bitcoin over Tor, which in fact creates an attack vector for deterministic and stealthy MITM attacks, as shown in [30].

Our results also suggest that our participants trust the cryptography behind Bitcoin and are aware of risks according to value fluctuation and software vulnerabilities. Poor usability and the lack of knowledge are major contributors to security failures. Almost a fourth of our participants indicated that they had already lost bitcoins or Bitcoin keys at least once (Q5). To our surprise, almost half of those who lost bitcoins due to a self-induced error which indicates that state of the art CMTs are sometimes still difficult to use or require users to manually take care of security tasks, such as backups and encryption. Our results also indicate that the Bitcoin ecosystem is mostly utilized for tipping and donations as well as acquiring digital goods, but to some extend also for criminal activity and adventurous gambling.

7.9 Conclusion

In this work we presented the first user study to examine how users interact with the Bitcoin ecosystem in terms of security and privacy. We conducted an online survey with 990 Bitcoin users and qualitative interviews with a subset of 10 participants. Furthermore, we introduced the term *Coin Management Tools (CMTs)* to describe tools that let users manage their virtual assets (keys) and interact with the Bitcoin network. Additionally, we proposed a method for categorizing CMTs according to the degree of control and verifiability a user can exercise with this client.

We found that managing bitcoins is still a major challenge for many users, as many of them do not apply sufficient security measures such as encryption and backups. We found that many participants were not even aware of security features provided by their used CMT. Two of the most widely used CMTs among our participants were web-hosted solutions. About half of their users reported to use such solutions exclusively, while the other half also used local clients. Even though web clients ought to be a usable and convenient solution, they require a certain level of trust and shift the responsibilities of encryption and managing backups to a third party. We also found that 22.5% of our participants have already experienced security breaches and lost bitcoins. About half of them mentioned a self-induced error as the reason, which highlights that users find it still difficult to manage their bitcoins in a secure way.

We believe that our insights and suggestions are an important first step towards improving the usability of Bitcoin security. In order to guarantee secure interactions with the Bitcoin ecosystem to both expert and non-expert users, we must re-think the concept of Bitcoin management, since it is more than just the secure handling of secret keys. Bitcoin is a decentralized system where the interactions between peers and the propagation and verification of messages and data is important. If this aspect is ignored, Bitcoin would just consist of signed numbers without value.

CHAPTER 8

TLS Usability from an Administrator's Perspective

Protecting communication content at scale is a difficult task, and TLS is one of the most common deployed protocols to do so. However, deploying it correctly has been shown to be challenging. In this chapter we present the findings on examining the usability of the TLS deployment process in HTTPS. We performed a series of experiments with 28 expert users and revealed significant usability challenges that result in weak TLS configurations. Additionally, we conducted expert interviews with seven experienced security auditors and penetration testers. Our results suggest that the deployment process is too complex even for experts and that server configurations should have strong security by default. We also found that even experienced users rely heavily on online resources to make major decisions (e.g., which cipher suites to use) as they do not have sufficient knowledge of the underlying cryptographic fundamentals. While the results from our expert interviews confirm the ecological validity of the lab study results, they additionally highlight that even experts prefer solutions that are easy to use, and that an improved and less vulnerable workflow would be beneficial to finding stronger configurations in the wild.

The remainder of this chapter is structured as follows: In Section 8.1we present an introduction and in Section 8.2 we discuss related work. In Section 8.3 we present our study methodology and in Section 8.4 the results of both the security and usability evaluation. Section 8.5 describes the methodology and results from our expert interviews. Section 8.6 discusses and Section 8.7 concludes our work.

8.1 Introduction

Transport Layer Security (TLS) is a fundamental cryptographic protocol to secure communications over the Internet and to ensure privacy and data integrity between two

communicating parties. Several versions and implementations are used in applications such as e-mail, chat infrastructures and major websites. Furthermore, there are multiple possible ways of deploying these protocols with severe implications on security. Also, there is a huge market of certificate authorities (CAs) that issue X.509 certificates used for authentication. Despite all efforts in propagating encryption, a large number of services and websites still refrain from using TLS or are poorly configured and, therefore, remain vulnerable to all kind of different attacks. Recent studies on TLS usage in the Internet ecosystem for both HTTPS [89, 58] and email [88, 132] revealed that most communication is poorly secured in transit and still susceptible to Man-in-the-Middle (MITM) attacks.

Most modern browsers support TLS and inform their users whether the communication between the browser and the service is secured properly. Whenever a user tries to access a website with an invalid certificate, a warning is issued by the browser. Human-centric studies[12, 65] have shown that these warnings are mostly clicked through and that users have little to no understanding regarding the implications of visiting a site without a valid certificate. In December 2015, the initiative *Let's Encrypt* released its non-profit CA that provides free domain-validated X.509 certificates and software to enable installation and maintenance of these certificates was launched to make it easier for administrators to deploy TLS.

To date, the studies on human-centric concerns focused on non-expert end users and, to the best of our knowledge, no user study has yet been conducted to examine the usability of the TLS deployment process. Our contribution aims to fill this gap by presenting the first user study with expert users to identify key usability issues in the deployment process of TLS that lead to insecure configurations. We conducted lab sessions that lasted 2 hours each with 28 participants from 14 to 18 December 2015. Data was collected via a think-aloud protocol as well as an entry and exit questionnaire. In addition we collected the bash and browser histories and the resulting server configuration files. We found that configuring TLS on Apache is perceived as a challenging task even by experienced users. Our results suggest that administrators struggle with important security decisions (e.g., choosing the right cipher suites) which are mainly driven by concerns about compatibility. Furthermore, our participants had a hard time finding reliable sources on the Internet to support their decision making process. The configuration options in Apache are perceived as difficult to understand and therefore an additional source of error. Through our expert interviews, we collected evidence that insufficiently secure configurations – like those from the majority of participants from our lab study – are frequently encountered in the wild and during security audits. Our results shed light on major challenges from an expert user's perspective. We are confident that our results are a good baseline for the development of improved tools and policies that are better tied to the expert users' needs.

The contributions of this chapter are:

- a **lab study with 28 expert participants** to explore usability challenges in the TLS configuration process
- **expert interviews with 7 security auditors** to provide a baseline for ecological validity and to further explore potential usability improvements of the deployment process
- a set of **recommendations** to ease the pain of securing communication content at scale.

8.2 Background

Transport Layer Security is the foundation of today's web security. Several application layer protocols use TLS to secure their online communication. The most widely used protocol is HTTPS, i.e., TLS provides confidentiality, authenticity and integrity for HTTP. Currently, TLS 1.2 [56] is the most recent version of the SSL/TLS protocol family, with TLS 1.3 on the horizon.[1] Besides securing the majority of today's web traffic, researchers have found several challenges regarding TLS, which are vigorously discussed in the literature [47, 167]. Guidelines and best practices for a proper TLS deployment have also been published [168, 39]. The goals of TLS include extensibility and interoperability. This includes the ability to change the quality of the used certificate, settings of used cryptographic primitives (cipher suites), enabling of TLS extensions, use of different TLS versions and the use of additional security features like HTTP Strict Transport Security (HSTS) [87] and HTTP Public Key Pinning (HPKP) [62]. In the last years, many studies focused on empirically testing the quality of TLS configurations by using Internet-wide scanning techniques and showed that the TLS landscape is diverse and full of misconfigurations. Lee et al. [124] analyzed the supported SSL/TLS versions, the EFF started to analyze used certificates [59, 60] with the latest and most comprehensive study by Durumeric et al. [58]. Ristic [156, 157] analyzed different parameters and evaluated the quality by a defined metric [9]. Huang et al. [92] surveyed the use of cipher suites and Kranch and Bonneau [109] scanned domains for HSTS and public key pinning.

8.2.1 Related Work

Most user studies regarding TLS and human-computer interaction focus on non-expert end users that receive certificate warnings from their browsers. Akhawe et al. [12] performed a large-scale study on the effectiveness of SSL browser warnings and found that that these warnings have high click-through rates, i.e., 70.2% of Google Chrome's SSL warnings did not prevent users from visiting the initially requested insecure site. Harbach et al. [78] presented an empirical analysis of the influence of linguistic properties

[1] `https://tools.ietf.org/html/draft-ietf-tls-tls13-12`

on the perceived difficulty of descriptive text in warning messages and found that the several steps can help to improve text understandability.

Several studies have been conducted to improve SSL warnings [65, 178, 188, 66]: E.g., Sunshine et al. [178] conducted a survey to examine Internet users' reactions to and understanding of current SSL warnings. Based on their findings, they designed new warnings and showed that they performed significantly better. Weber et al. [188] used a participatory design approach to improve SSL warnings. Felt et al. [66] explored reasons for higher click-through rates for SSL warnings in Google Chrome compared to Mozilla Firefox. They also showed that the design of warnings can lead users towards safer decisions.

Oltrogge et al. [144] conducted an extensive study on the applicability of pinning for non-browser software as in Android apps. They found that only a quarter of their participants understood the concept of pinning. Based on their findings, they presented a web application to support developers in making the right decisions and guiding them through the correct deployment.

8.3 Lab Experiments

In the following, we describe the methodology used to collect and analyze the data from the lab study.

8.3.1 Study Design and Procedure

In order to elicit a picture of usability challenges of TLS deployment from an administrator's point of view, we conducted a series of lab experiments with 28 participants. As described in Section 8.3.2, we recruited participants with expert knowledge in the field of security and privacy-enhancing protocols at our university who fulfilled the criteria to potentially work as an administrator or were actually working as administrators.

Our experiments proceeded as follows: After the recruitment phase, the participants were invited to the lab where they were shortly briefed about the purpose of our study. After signing a consent form, they received the assignment. In the scenario, they assumed the role of an administrator of an SME who is in charge of securing the communication to an Apache web server with HTTPS in order to pass a security audit. We prepared and implemented a fictive Certificate Authority (CA) in order to facilitate the process of getting a valid certificate and to remove any bias introduced by the procedures from a certain CA. The browser on the local machine already trusted our CA.

We instructed the participants to make the configuration as secure as possible, whereas the assignment did not contain any specific security requirements, such as which cipher suites to use or whether to deploy HSTS or not. In order to collect data, we used a

think-aloud protocol. While the participants where working on the task, they articulated their thoughts while an experimenter seated next to them observed their work and took notes. We refrained from video recording due to the results from our pre-test during which we filmed the sessions and noticed a severe impact on the participants' behavior. The participants from the pre-study also explicitly reported that they perceived the cameras as disruptive and distracting, even though they were placed in a discreet way. Figure 8.1 shows the experimental setup in the lab.

In addition to the notes from the observation, we captured the bash and browser history and the final configuration files. After completing the task, the participants were asked to fill out a short questionnaire with closed- and open-ended questions which covered basic demographics, previous security experience in industry and reflections on the experiment. The complete assignment and questionnaire can be found in the Appendix of this paper.

As a result, we had a collection of both qualitative and quantitative data that was further used for analysis as described in Section 8.3.3.

8.3.2 Recruitment and Participants

In contrast to previous studies in the area of TLS usability, we focused on expert users that have proficient knowledge in the field of security and privacy-enhancing technologies. As it was very difficult to recruit participants from companies, irrespective of a financial incentive, we decided to recruit participants at the university and targeted students that had previously completed a set of security courses. We invited a selected set of students to participate in an online quiz to additionally assess their knowledge irrespective of their previously issued grades. The quiz covered knowledge questions about Linux skills, web security and encryption. The top 30 students with the best scores were then invited to participate in the lab study, and 28 of them did. Table 8.1 summarizes key characteristics of the participants: 3 participants were female, 27 were male; the age range was 21 to 32 with a median of 23. Their experience working in industry ranged from 2 to 120 months with a median of 25 months. 18 of our 28 participants were already experienced system administrators and reported to have deployed TLS before.

8.3.3 Data Analysis

For a qualitative analysis of the observation protocols we followed the grounded theory methodology of Strauss and Corbin[177] which is often used in usable security research to develop models and theories from qualitative data, e.g., [176, 149, 104]. The grounded theory approach involves several steps in the analysis process and was implemented as follows: At first, two researchers traversed all data segments independently point-by-point and assigned descriptive codes. This process is referred to as *open coding*. The two researchers performed the initial coding independently from each other to minimize the susceptibility of biased interpretation. We evaluated the quality of our initial codes

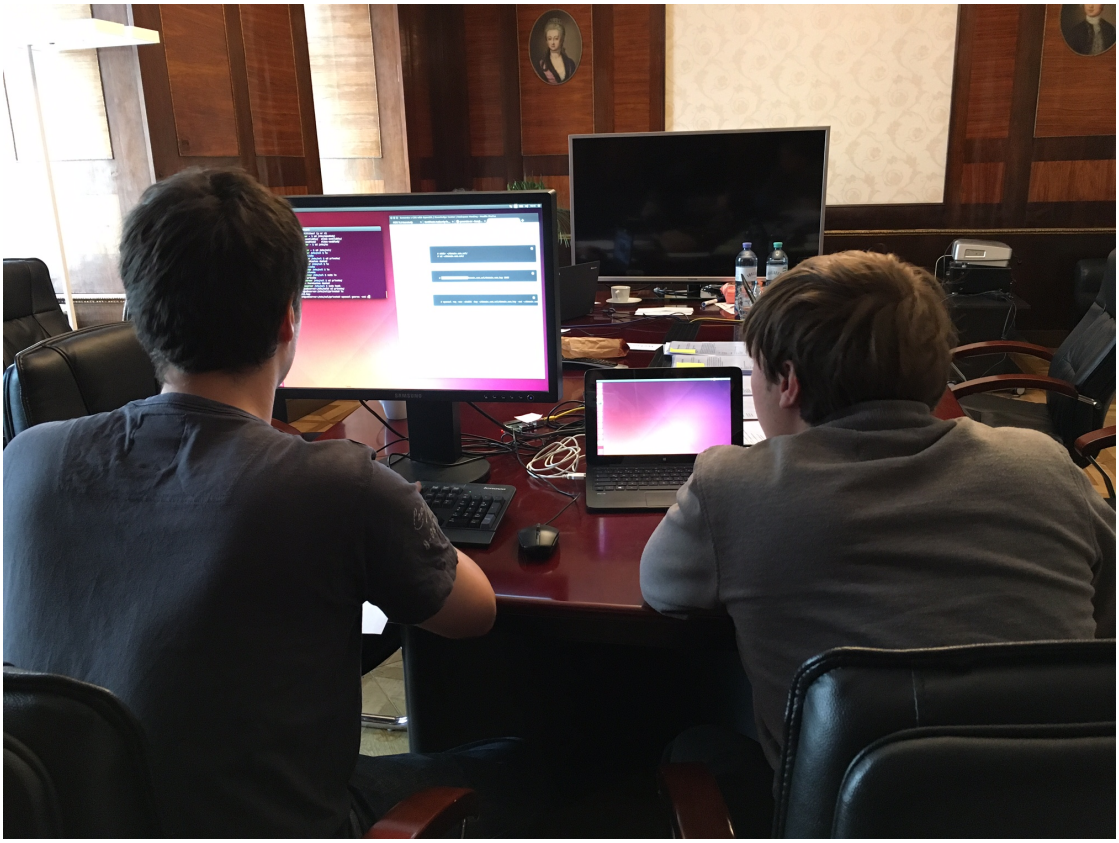


Figure 8.1: Experimental setup.

and found a good inter-coder agreement between the two researchers (κ=0.78). On the resulting initial set of coded data we performed *axial coding* to look for explanations and relationships among the codes and topics to uncover structures in the data. Then we performed *selective coding* to put the results together and derive a theory from the data.

In order to structure the data from the open-ended questions collected through the questionnaire we used an *iterative coding* process. Hence we went through the collected data and produced an initial set of codes. Then we revised the retrieved codes and discussed recurring themes, patterns and interconnections. After agreeing on a final set of codes, we coded the entire data. As a result, we obtained a picture of usability challenges in the deployment process which is presented in Section 8.4, grouped by themes.

To evaluate the (mostly) quantitative data acquired via the bash/browser history and Apache log files, we applied metrics and measures to evaluate the quality of the resulting configuration, as defined in 8.4.1.

Table 8.1: Participant characteristics from the lab experiments. n=30

Demographic	Number	Percent
Gender		
Female	3	10%
Male	27	90%
Age		
Min.	21	
Max.	32	
Median	23	
Months worked in industry		
Min.	2	
Max.	120	
Median	25	
Experienced as sysadmin		
Yes	18	60%
No	12	40%
Configured TLS before		
Yes	18	60%
No	12	40%
Currently administrating		
Company web server	5	17%
Private web server	17	83%

8.4 Results

In this section we present the results from our lab study which are based on the data from the think-aloud protocol, the collected log files and the self-reported data from the exit-questionnaire.

8.4.1 Security Evaluation

We based our evaluation criteria on Qualy's SSL Test[2]. We consider this rating scheme a useful benchmark to assess the quality of a TLS configuration based on the state of the art recommendations from various RFCs [168, 167] and with respect to the most recently discovered vulnerabilities and attacks in the protocol. It must be mentioned that this benchmark reflects the best-case scenario at the time of writing, but could be different in the future if new vulnerabilities are discovered.

[2]https://www.ssllabs.com

Table 8.2 summarizes the results of a security evaluation based on the final configuration per participant with additional information in Table 8.3. Section 8.4.1 briefly describes our evaluation criteria based on the metrics used in Qualy's SSL Test.

Only four participants managed to deploy an A grade TLS configuration, P24 received the best overall score. B was the most commonly awarded grade (15 out of 28). Four participants did not manage to deploy a valid TLS configuration in the given time (P7, P18, P23, P26). Two participants (P10 and P19) encrypted their private keys. One of them did not share the passphrase with us. However, the passphrase was easy to brute-force.
Fortunately, none of our participants chose a key size smaller than 2048 for their RSA key. 15 participants chose 2k- and eight chose 4k-sized keys. Five out of the 28 participants deployed the certificate chain correctly, which is necessary to receive a grade better than B according to our rating scheme.
Two participants did not make use of the study CA and used self-signed certificates. Only one participant enabled a TLS version lower than TLS 1.0 (P8), another participant had all versions but TLS 1.2 disabled (P14). Only two participants configured RC4 support and only one configuration (P8) was vulnerable to the POODLE attack as SSL 3 was still supported. 14 participants fully configured forward secrecy, the remaining participants with valid configurations managed to at least partially support it. Eleven participants included HSTS headers to improve the security of their configuration and only two participants deployed HPKP.

Evaluation Criteria Our evaluation is based on the evaluation criteria from Qualys SSL Labs [9]. The rating is expressed with a grade from A to F and composed out of three independent values: (1) protocol support (30%), (2) key exchange (30%) and (3) cipher strength (40%). Some properties, e.g., support for the RC4 cipher cap the overall grade as shown in Table 8.3. For improved readability, the detailed criteria is presented in the Appendix.

8.4.2 TLS Deployment Model

Our grounded-theory-based analysis of the think-aloud protocols from our lab study yielded a process model for a successful TLS configuration. All participants who managed a valid configuration in the given time can be mapped to the stages presented in this model. The four participants who did not manage to deploy TLS in the given time significantly deviate from this model. We divide the steps from our model into two phases, a *setup phase* and a *hardening phase*. We refer to the *setup phase* as to a set of tasks to get a basic TLS configuration, i.e., the service is reachable via https if requested. The *hardening phase* comprises all necessary tasks to get a configuration which is widely considered *secure* with respect to the metrics defined in 8.4.1. Figure 8.2 shows our deployment model. Participants who achieved at least a basic configuration successfully completed all steps of the setup phase, while better-graded configurations completed

ID	Grade with Trust Issues Ignored	Errors / Warnings / Highlights	Cipher Strength Score	Key Exchange Score	Protocol Support Score	Common Name	Key Size	Certificate Chain Length	Used Provided CA to Sign	Encrypted Private Key	SSL 2	SSL 3	TLS 1.0	TLS 1.1	TLS 1.2	RC4 Support	Vulnerable to POODLE (SSL 3)	Forward Secrecy	HSTS	HPKP
P1	A	2	90	90	95	web.local	4096	3	●	○	○	○	●	●	●	○	○	●	●	○
P2	B	3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	○	○
P3	B	2,3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	●	○
P4	A		90	90	95	web.local	2048	3	●	○	○	○	●	●	●	○	○	●	○	○
P5	B		90	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	●	●	○
P6	B	3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	○	○
P7	Not valid																			
P8	C	3-6,8	90	90	50	web.local	2048	1	●	○	○	●	●	○	○	●	●	◐	○	○
P9	B	1-3	100	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	●	●	●
P10	B	1-3	90	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	●	●	●
P11	B	3,4	90	90	95	web.local	2048	1	●	●	○	○	●	●	●	○	○	◐	○	○
P12	B	2,3	90	90	95	web.local	4096	1	●	○	○	○	●	○	●	○	○	●	●	○
P13	B	3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	◐	○	○
P14	A-	4	90	90	100	raspberrypi	2048	1	○	○	○	○	○	○	●	○	○	◐	○	○
P15	C	4,7	50	90	95	-	2048	1	○	○	○	○	●	●	●	●	○	◐	○	○
P16	A-	4	90	90	95	web.local	2048	3	●	○	○	○	●	●	●	○	○	◐	○	○
P17	B	2,3	90	90	95	web.local	3096	1	●	○	○	○	●	●	●	○	○	●	●	○
P18	Not valid																			
P19	B	2,3	90	90	95	web.local	2048	1	●	●	○	○	●	●	●	○	○	●	●	○
P20	B	2,3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	●	○
P21	B	3,4	90	90	95	Test	2048	1	●	○	○	○	●	●	●	○	○	◐	○	○
P22	B	3,4	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	◐	○	○
P23	Not valid																			
P24	A	2	90	90	97	web.local	2048	3	●	○	○	○	○	●	●	○	○	●	●	○
P25	B	3	90	90	95	SME	4096	1	●	○	○	○	●	●	●	○	○	◐	○	○
P26	Not valid																			
P27	B	3,4	90	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	◐	○	○
P28	A	2	90	90	95	web.local	4096	3	●	○	○	○	●	●	●	○	○	●	●	○

Table 8.2: Security evaluation of the final TLS configuration per participant.

some steps from the hardening phase as well. We identified iterative (tool-supported) security testing as a key element for a successful hardening phase, since the participants relied on external sources to evaluate the quality of their configuration.

8.4.3 Usability Challenges in TLS Deployment

In the following, we present the usability challenges identified through our grounded theory analysis of qualitative data from the think-aloud protocols and the quantitative data from the collected log files.

1	Highlight	HTTP Public Key Pinning (HPKP) deployed on this server. Yay!
2	Highlight	HTTP Strict Transport Security (HSTS) with long duration deployed on this server.
3	Warning	This server's certificate chain is incomplete. Grade capped to B.
4	Warning	The server does not support Forward Secrecy with the reference browsers.
5	Warning	This server accepts RC4 cipher, but only with older protocol versions. Grade capped to B.
6	Warning	The server supports only older protocols, but not the current best TLS 1.2. Grade capped to C.
7	Warning	This server uses RC4 with modern protocols. Grade capped to C.
8	Error	This server is vulnerable to the POODLE attack. If possible, disable SSL 3 to mitigate. Grade capped to C.

Table 8.3: Errors / Highlights / Warnings as referred to in Table 8.2.

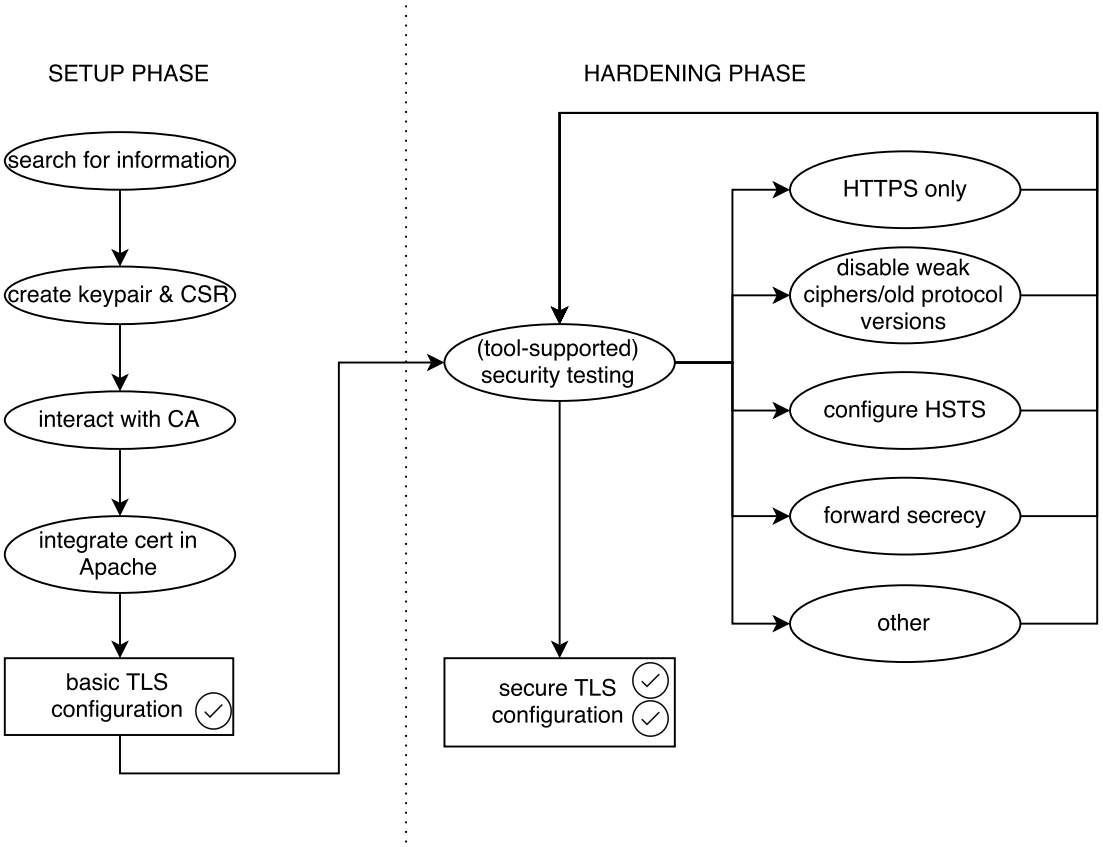


Figure 8.2: Schematic representation of a successful workflow.

Searching for information and finding the right workflow Except for a handful of very experienced participants, who explicitly searched for tutorials they were aware of, the study participants visited a high number of websites and used multiple sources of information. The information sources were very diverse regarding their suggested deployment approaches and information quality respectively. We frequently observed that a participant started to follow an approach from one tutorial and soon had to switch to another as the presented approach was not feasible for our deployment scenario and

the given server configuration. The lowest number of visited websites during the lab study was 19 (P21). In contrary, participant P4 visited 137 websites during the given time. The average number of visited websites over all participants was 60 (median=49.5, sd=27). We consider this a relatively high number given the low amount of time.

Most participants expressed annoyance and vexation about the incompatibility of the different information sources. We also found that the number of visited websites (high, medium, low) does not impact the quality of the resulting configuration, but this result is not significant in our sample
with $\chi^2(0.23327892, 6) > 0.05$.

Table 8.4: Participants and their cumulative number of visited sites and overall rating.

Participant ID	**Visited websites**	**Grade**
Most visited sites		
P4	147	A
P19	116	B
P8	111	C
P2	109	B
P7	116	-
Least visited sites		
P21	20	B
P12	36	B
P5	49	B
P10	49	B
P18	50	-

Table 8.5: Top most visited websites.

URL	**Visitors**
`wiki.ubuntuusers.de/Apache/SSL`	25
`httpd.apache.org/docs/2.4/ssl/`	20
`www.ssllabs.com/`	16
`bettercrypto.org`	15
`raymii.org/s/tutorials/Strong_SSL_Security_..`	14
`httpd.apache.org/docs/2.2/mod/mod_ssl`	11

Creating a Certificate Signing Request (CSR) A CSR is a block of encrypted text which is sent to a CA to request a TLS certificate. It therefore contains information that will be included in the certificate such as organization name and common name (FQDN) and enables users to send their public key along with some information that

identifies the domain name in a standardized way. When creating a CSR, the user is asked to fill out the respective information. In order to create a CSR, the user has to create a key pair. Our results suggest that many users do not understand the purpose and concept of a CSR, i.e., who it is authenticating towards whom. 19 out of 30 participants from the lab study had to create two or more requests due to errors in the CSR creation. The most common error was that they did not fill out the requested common name field correctly (14 participants) and thus did not receive a valid certificate for their domain. In the end, 20 participants created a CSR with the correct common name as shown in Table 8.2. As this is a common error in practice, some CAs even highlight that the common name(s) can be altered later on. This is especially useful when adding TLS support for subdomains. Second, two participants (P14 and P15) did not fully understand the difference between a CSR and a (self-signed) certificate. Six participants initially created a self-signed certificate instead of a CSR and tried to upload it to the CA. According to the self-reported work experience, this happened to participants regardless of their experience. E.g., P15 reported to have recently deployed TLS on Apache and still tried to upload a self-signed certificate to the CA. Four participants recognized the error after receiving an error message from the CA and then created a correct CSR including a correct common name.

Choosing the appropriate cipher suites In TLS, cipher suites are used to determine how secure communication takes place. Cipher suites are composed from building blocks in order to achieve security through diversity. A person in charge of configuring TLS has to select cipher suites that provide authentication and encryption that is considered strong. However, this is a task that requires a deep and up-to-date understanding on the underlying algorithms in order to make informed decisions about which cipher suites to support. In the course of our lab experiments, all participants who came to this point during the configuration assignment were aware of the fact that they had to manually select cipher suites to secure the communication. The decision making process was exclusively based on search results and suggestions from online resources without questioning. Some participants also referred to recently published blog posts were they read about the disadvantages of a certain algorithm. This implies that the quality of the used information source is crucial for the overall security of the configuration as our participants lacked profound knowledge and thus had to trust their source of information. Table 8.2 shows how the selected cipher suites impact the quality of the configuration.

Strict HTTPS After finishing an initial valid configuration, most participants enforced strict HTTPS as a first step of the hardening phase. Some were annoyed by the fact that HTTPS does not immediately replace HTTP as soon as it is available. Most participants were initially confused when they tested their configuration via the browser and were redirected via http when they entered the URL without the http(s):// prefix. They then spent a significant amount of time to configure the virtual host and the respective ports correctly, mostly also due to misleading or incomplete information from online sources.

Multiple configuration files All but six participants said that they found the configuration file structure confusing, regardless of their prior experience with Apache. E.g., P14 found it particularly challenging to find the right configuration files. According to the think-aloud protocol, this was the main challenge that in the end resulted in an invalid configuration. Several participants copied and pasted entries between different configuration files or had double entries, e.g., for SSLEngineOn. Nine participants also struggled with loading the modules, e.g., P18 did not understand where to load the modules in the configuration. Many participants were also not aware of where and how to create a new virtual host which listens on 443. P23 for example did not understand the differences between the http.conf and apache.conf which distracted him/her from the TLS-specific tasks and security-critical decisions.

Finding the right balance between security and compatibility We observed that the majority of our participants struggled with the definition of a *secure* configuration. In our assignment we just stated that the configuration should be *as secure as possible to withstand an audit*, without specifying any key properties. Hence, the participants themselves had to make the decisions. About 15 participants expressed concerns regarding compatibility when configuring SSL/TLS versions and cipher suites. A majority of them, however, decided in favor of the securer option, e.g., disabling all TLS versions < TLS v1.1 and thus refraining from supporting older versions of IE.

8.4.4 Perceptions of Usability

After the lab experiments, the study participants filled out a short online questionnaire and reported reflections on the assignment. 18 participants reported that they thought they finished the assignment completely, while nine thought that there were still some configuration steps missing. One participant was not sure about whether or not he/she finished the task. While ten participants perceived the assigned task as difficult and three as very difficult, only four participants thought that it was easy and one that it was very easy. Twelve participants rated the difficulty as neutral.

We also asked our participants what they think are the most severe usability pitfalls in the deployment process. In the following, we provide a respective list. Most frequently mentioned were lack of best practice tutorials (19), followed by misleading terminology (15) and weak default configurations (12).

Lack of reliable information According to our participants, it was very hard to determine a best practice on how to deploy TLS. Our participants reported that they came across outdated or simply wrong information in online tutorials. 13 participants also mentioned that most tutorials were not generic, but still not specific enough to apply them to the system given in the assignment.

Misleading terminology and error messages Especially with respect to interactions with the CA, participants expressed confusion about the terminology. Some

accidentally uploaded a self-signed certificate instead of a CSR and found the file endings difficult to handle and to distinguish, e.g., *.key*, *.pem*, *.crt*.

Weak default configuration Eight participants explicitly criticized the high effort necessary to harden the configuration, as too many cipher suites are enabled by default. Also, they criticized that the selection of cipher suites is a time-consuming task that requires profound background knowledge in order to make an informed decision and that bad decisions yield major security vulnerabilities. One participant also suggested a simplified configuration option including a two- or three-way variable to disable certain cipher suites (e.g., tinfoil hat vs. maximum compatibility). Four participants also stated that they would prefer if web servers had TLS configured by default.

Confusing config file structure During the configuration process, many participants perceived the Apache config file structure as confusing and experienced it as a severe source for errors. We also observed that some participants had simple copy/paste errors in their config files which highly distracted them from the actual main task.

Complex workflow Six participants explicitly stated that the workflow itself is too complex due to the different approaches and branches that can be taken during the configuration process as well as the dependencies of the subtasks. Three participants stated these factors hindered them in finding the source of an error afterwards.

Too much background knowledge required Many participants expressed their concern about the high amount of background knowledge required to successfully configure TLS in a secure way. Also, the fact that a TLS configuration must be well maintained and frequently updated requires the person in charge to be informed about the latest TLS attacks and other vulnerabilities which our participants considered infeasible in practice.

Confusing permissions Five participants also stated that they found it hard to choose the correct location and permissions for the certificate and private key.

Participant Statements

- *"It seems that there is already a certificate called snakeoil, why can't I use this one?"* (P7)
- *"The configuration process is fiddly and one has to google tons of pages to get it right. Even then one cannot be sure to have a good configuration because SSL vulnerabilities are discovered almost on a regular basis."* (P9)
- *"There are multiple config files in /etc/apache2, how and where do I have to load modules?"* (P18)
- *"Why is there a snakeoil certificate in the config file?"* (P22)

- *"I have absolutely no idea what I'm doing. Neither am I aware of whether my online source is trustworthy."* (P23)

8.5 Expert Interviews

In order to address ecological validity, we conducted additional expert interviews with security consultants and auditors about their experiences with insecure TLS configurations. In this section, we describe the interview methodology and results of these expert interviews that were conducted in April 2016.

8.5.1 Interview Procedure

The expert interviews were conducted as semi-structured interviews with ten security experts from well-respected security consulting firms in the German-speaking region. The experts were familiar with TLS misconfigurations and frequently encountered misconceptions on how to combat the trade-off between compatibility and security. The interview segments were coded using iterative coding.

8.5.2 Results

Our results show that auditors commonly agree that poor usability and too complex workflows and server configurations result in weak TLS configurations. They also mentioned that the deployment process must be simplified and especially the default configuration should favor security. In the following, we will discuss their responses in detail. Six interview participants were male, one was female. The average number of months spent as a penetration tester or auditor was 53.2. Two participants work in a very small company with < 10 employees, the remaining participants work in a company with more than ten but less than 100 employees.

Auditing TLS configurations All expert interview participants focus on the following configuration characteristics: activated TLS/SSL version, activated cipher suites, if the certificate is recognized by commonly used web browsers, whether HSTS is configured and whether public key pinning is activated. E3 and E7 also highlighted that they particularly pay attention if recently discovered attacks are mitigated. E6 and E7 also said that in addition to automated tools, they like to evaluate the server configuration directly, if it is accessible.
All seven interview participants use Qualy's SSL Test as the de-facto standard to evaluate public domains. According to them, they preferably use Qualy's SSL Test, but they also use selected Nessus modules[3] and OpenVAS[4] for internal sites. E2, E4 and E6 also reported to use NMap[126].

[3] http://www.tenable.com/products/nessus-vulnerability-scanner
[4] http://www.openvas.org/

Configuration mistakes in the wild According to the interview participants, the main concern when deploying TLS is compatibility. Our interviewees, however, also mentioned that in most cases the compatibility challenge is just a mock argument which is often used as an excuse and not fully elaborated by the responsible employees. Compatibility is a challenge for publicly available sites where almost any client may want to access. However, it is a rather easy-to-solve problem for services that are only accessed internally, hence the set of potentially accessing clients is well known. Also, backward compatibility with older client versions (i.e., < IE7) may not be desired for a variety of reasons beyond TLS and will only affect a minority of clients. However, E1 and E3 also reported that finding the best fit between security and compatibility is hard even for security experts and often arguable. Five of the interviewed auditors also reported that they often find self-signed certificates which do not fulfil the intended purpose. E1, E2, E3 and E7 also mentioned that they often encounter weak default TLS configurations with poor ciphers and no additional security measures (e.g., HSTS).

Two auditors mentioned that in the course of looking at TLS configurations for many years, they have never encountered HTTP public key pinning during an audit. Also, one interview participant reported that TLS deployment is not sufficiently streamlined in companies. According to them, most companies have multiple servers with varying configurations and each one is maintained and updated separately.

E2 also highlights that the ideal TLS configuration has changed frequently in the last two years, i.e., new attacks have been published (e.g., Heartbleed [11] and DROWN [17]) or algorithms have been deprecated which implies a significant overhead for administrators to keep their configurations up to date. E2, E4 and E7 also reported that companies do not fully make use of the online sources available, e.g., using Qualy's SSL test for public domains.

"In most cases backward compatibility is the show-stopper regarding proper TLS configurations." (E3)

Concerns in the wild We also asked our interviewees about the concerns that admins, CSOs and other persons in charge have regarding TLS. Our experts agreed that especially administrators are aware that configuring TLS is a sensitive task during which several things can go wrong. However, lack of time seems to be a major issue and administrators often do not have the resources to get a deep understanding on the fundamentals. To our surprise, E4 and E7 reported that they frequently encounter responsible persons that have little or no experience with security protocols such as TLS. All interview participants reported that in the course of security audits, they also frequently find weak default configurations along with very little awareness regarding the weakness of such configurations and how they could easily be hardened. E7 highlighted that responsible persons even report that they are "afraid of using crypto". As an example (described

in 8.5.2), E1 explicitly mentioned HSTS which is easy to deploy and has no impact on compatibility, but is rarely used in practice.

Also, compatibility still remains a key concern as lack of compatibility often leads to overloaded help lines, as reported by E1, E6 and E7. Also, the risk of MITM attacks is often underestimated and companies do not perceive themselves as targets of such attacks. E7 cited an administrator from an SME saying: "Our configuration supports basic encryption, so this should be more than enough... and clearly is better than no encryption." As E2 reports, companies are often concerned about introducing encryption due to the additional performance overhead which is in their opinion not worth the effort.

Suggested usability improvements A common opinion of all interviewees was that the default server configurations must be improved by simplifications and default security options. They said that server configurations should be secure by default, i.e., that TLS should be enabled by default and hence must be explicitly disabled if necessary. E1 highlighted that Apache has a weak default configuration for compatibility reasons and mentioned the Caddy web server[5] as a good and usable example. Caddy comes with a TLS configuration by default and uses *Let's Encrypt* to get certificates. Also, according to E1 the by default activated cipher suites are a good compromise, and even OCSP stapling and HSTS are deployed by default. Also, the Caddy web server automatically renews certificates. E1 highlights that configuration directives must be simplified to yield strong configurations and that Caddy web server is a good example for this paradigm. E1 also suggests that compatibility flags which administrators can use to configure cipher suites would be much more helpful than letting them deal with cipher suites directly.

Regarding the deployment process in larger enterprises that maintain multiple servers, E1 proposes to create a strong sample configuration on a test server and to then deploy them on all servers. This potentially helps to avoid outdated configurations, as the updating process is simplified and the person in charge is aware of the TLS configuration on all devices by knowing the essentials of the sample configuration.

E1 also suggests to deploy everything that does not result in lower compatibility, i.e., OCSP stapling which is commonly ignored by clients who do not understand the according header. While public key pinning is rather difficult to fully deploy, it can easily be used in report-only mode and thus enable to detect MITM attacks. E1 highlights that these additional functionalities are beneficial for security but rarely encountered in the wild.

E3 also suggests that HTTPS should fully replace HTTP to solve security problems. E3 also thinks that HTTP has no fundamental benefit over HTTPS with TLS. E3 also plays the ball from servers to clients and says that clients should be frequently updated to support the respective ciphers. Furthermore, E3 argues that the concept behind CAs also has its flaws, i.e., lack of certificate transparency, certificate revocation and lawful interception on the CA's side without the end user's consent. She/he also claims that browsers generally trust a high number of CAs with varying trustworthiness.

[5] `https://caddyserver.com/`

E7 highlighted the need for professional education and that "doing it right" requires experienced professionals that keep track of the ongoing changes. E7 also suggested that there is a high demand for better configuration guides and easier-to-use default configurations to compensate the lack of know-how of the persons in charge as well as to make it easier for everyone to configure TLS in a secure yet compatible way. Also, this interview participant said that companies should have policies regarding encryption and compatibility to make it easier for administrators to choose the right configuration.

8.6 Discussion

While related work already showed that TLS configurations in the wild are often weak and thus do not sufficiently protect Internet users from MITM attacks, our work explores the reasons for this. In comparison to related user studies, we focus on the expert user role instead of the non-expert end user who is mostly unaware of potential risks and clicks through warnings which are often hard to understand and do not sufficiently communicate security risks.

We were surprised by the helplessness that we encountered during the lab study. The security auditors who participated in our expert interviews draw a similar picture of the administrators' reaction when confronted with the results of an audit.

Our results suggest that poor usability is a key issue and by far the main reason for weak configurations. Through both our lab study and the expert interviews we found that even professionals lack the knowledge regarding the underlying cryptographic fundamentals such as cipher suites and even basic concepts like the role of certificates. This result shows that there is a high demand for better default configurations and/or tool support to prevent administrators from dealing with mechanisms they cannot fully understand. As mentioned by our security experts, there are already servers with a focus on better security: they let their users make configurations less secure if desired instead of providing no security by default and thus forcing users to deploy security themselves. Also, they highlight the demand for easier user interfaces for configuration purposes. Our results also suggest that expert users are often unable to decide on the appropriate level of security, which highlights the need for cross-organizational guidelines and policies.

As creating a basic TLS configuration also involves complex decisions (such as choosing the appropriate key length) it is very difficult for administrators to maintain them resp. correct errors or wrong decisions.

Both the results from the lab study and the expert interviews highlight that the complex deployment process should be simplified, and that the difference between a basic correct configuration and a secure one should not be too broad. Hence we suggest that newly designed servers and/or supportive tools should merge the setup and the hardening phase resulting in a best-case working configuration if all steps are completed – which can then be downgraded if necessary.

8.6.1 Limitations

A severe limitation of our lab study is that we only looked at the initial deployment process and excluded long-term maintenance effects, such as certificate renewal and the administrators' reactions to newly discovered vulnerabilities. The main reason is that it is difficult to reliably study long-term effects in the lab. In the future, we plan to conduct an additional case study in a corporate environment to observe long-term effects over a number of years. Also, as our study was performed in the lab, the participants did not have a deep background of the notional company they were administrating for the study. Our primary goal was to recruit participants who were fully employed as system administrators, but unfortunately did not manage to get enough responses resp. commitments for participation. Therefore, we chose to recruit participants among our computer science students. To overcome this bias, we selected top students that successfully completed security courses with good grades and completed an initial assessment test. As our results suggest, many of them were already experienced with managing servers and some had even worked as system administrators in companies and other organizations. We therefore believe that our data is suited to explore usability challenges. Our expert interviews with security auditors underline the ecological validity of the results from our lab study and suggest that configurations found in the wild are even less secure than those generated by our participants during the lab study.

8.7 Conclusion

We conducted a lab study with 28 participants to explore usability challenges in the TLS deployment process that lead to insecure configurations. In comparison to related work, we contributed a study that focuses on expert users, i.e., administrators who are in charge of securing servers. Additionally, we conducted seven expert interviews with penetration testers and security auditors who frequently encounter poorly secured servers during security audits.

We found that the TLS deployment process consists of multiple critical steps which, if not done correctly, lead to insecure communications and put Internet users at risk for MITM attacks. Furthermore, our results suggest that even computer scientists who are educated in terms of privacy-enhancing protocols and information security need additional support to make informed security decisions and lack an in-depth understanding of the underlying cryptographic fundamentals. Expert users also struggle with the configuration file structure of Apache web servers and have to put a lot of additional effort into securing default configurations. Our expert interviews underline the ecological validity of the results from our lab study and shed light on the weaknesses of TLS configurations found in the wild. According to our security auditors, the main concern regarding TLS is interoperability. They also highlighted that server infrastructures are often configured with poor defaults and badly maintained and are therefore not up-to-date.

CHAPTER 9

Summary and Future Work

In this thesis, we researched usable security and privacy in the context of technologies and tools which are currently considered as disruptive. These technologies pose significant challenges for users to manage their privacy and security in an always-online environment where information is continuously collected and shared.

We systematized social engineering attacks with respect to newly deployed technologies and novel attack vectors. Then we examined where stolen sensitive data (often through social engineering) is traded and presented a classifier to determine whether an online communication channel is used as an underground marketplace.

Furthermore, we addressed usable security and privacy challenges in specific application scenarios, with an emphasis on mobile and wearable computing. QR codes are often used to make information easily accessible on mobile and wearable devices. As they are non-human-readable, users rely on technology to determine whether they are under an attack. We therefore performed a series of experiments to determine vulnerabilities and to propose solutions to mitigate QR code-based attacks.

Another challenge in a wearable environment is bystander privacy. As there are no off-the-shelf privacy-mediating solutions available to date, we explored design directions for future solutions and proposed conceptual form factors that ensure accessibility and interoperability. As future work, we plan to further evaluate our concepts and to implement a prototype in order to conduct a longitudinal field study. The findings from our first field study highlight that there is a high societal demand for such a privacy-mediating system as individuals perceive recording in public without consent as a violation of their privacy. The fast proliferation of smart environments where devices continuously monitor their surroundings and the Internet of Things will make the situation even worse. In such scenarios not only graphical material will be recorded but various other sensors will be deployed to continuously collect information. Therefore, we identify usable privacy in smart environments as a major challenge for future research.

As future work, we plan to expand our research towards bystanders of non-graphical data collection in smart environments.

As mobile devices store a large quantity of sensitive information, they must be protected from unauthorized access. As users often choose weak secrets, we proposed an enhanced PIN scheme that assigns each digit a binary pressure value to make users select stronger secrets. Our evaluation through two user studies has shown that our system lets users select stronger PINs with only minimal impact on usability. Due to the scarce adoption of compatible hardware at the time the study was conducted and the results from a pre-study, we opted for a binary pressure scale. During our lab study, we measured the exact pressure values registered on the screen when users entered digits with force. These measurements suggest that there is the potential for a three-step pressure scale. Such a system would enable users to select even stronger PINs in theory. In practice however, it is necessary to study whether such three-step force-PINs are harder to remember and to what extent the task overhead increases. Therefore, as future work, we plan to expand our system and conduct user studies both in the lab and in the field to maximize the security benefit while keeping the impact on task overhead reasonably low.To do so, we will adjust our implementation to the newly determined thresholds and repeat our user studies respectively. Furthermore, we plan to conduct studies with a more heterogeneous sample to ensure that are results are generalizable to the entire population of smartphone users and to collect evidence on how participants from marginalized groups interact with the system. This is especially necessary to ensure accessibility of the system. We furthermore plan to develop and evaluate a similar system for Android devices with unlock patterns and to evaluate the usability and security respectively. We also suggest to consider the integration of pressure-sensitive components in other applications and devices that now rely on simple digit PINs.

Additionally, we emphasized on the usability of two cryptographic applications, namely the crypto-currency Bitcoin and Transport Layer Security (TLS).

In the field of Bitcoin, we conducted a large-scale user study to understand key management and other security and privacy challenges. Our results suggest that Bitcoin users do not sufficiently backup their digital assets and therefore do not manage to recover their lost keys and coins. Our results also indicate that many Bitcoin users suffer from such a loss which highlights the need for more usable tools to protect and backup bitcoins. As users of some wallets or Coin Management Tools (CMTs) make more backups compared to others, it is worth to conduct a more thorough analysis on usability factors that motivate users to do so. Another interesting finding from our study is that users favor online wallets. This behavior is paradoxical as such centralized services are contradicting the decentralized nature of Bitcoin. We are currently working on a systematization and a set of evaluation criteria for CMTs to aggregate the characteristics, benefits and drawbacks of the available tools. Based on these findings we plan to design a new CMT which combines the usability and security benefits. To aggregate our findings and to put it in line with our ongoing research in this area we are also assembling a book on crypto-currencies with an emphasis on Bitcoin. In the course of our research

related to Bitcoin, we also found that many users do not understand the underlying cryptographic concepts. Furthermore, the tools they interact with use metaphors that create mental models which do not fully reflect the underlying functionality e.g., the "coin" metaphor suggests that users hold coins. In reality however, users do not possess coins but cryptographic keys instead. We therefore plan to work on better metaphors for Bitcoin tools. As a first step towards improved metaphors and according design guidelines for CMTs, we plan to research user mental models of the crypto-currency through (mostly qualitative) lab studies. We furthermore hope that the concepts gathered through this research are helpful for the design of other cryptographic applications for e.g. message encryption.

Regarding TLS, we performed a user study with expert users to determine usability challenges during the deployment process. In the course of this study, we identified major usability flaws during the deployment process of TLS and showed that even expert users struggle when trying to configure TLS in the most secure way. This is not only due to the design of server configuration files but also due to the complexity of the underlying fundamentals which are frequently subject to change as new vulnerabilities and breaches are frequently discovered. Our results bring a completely new perspective on TLS security research as our study is the first one that focusses on vulnerabilities introduced by the person who configures the system. Previous studies focussed on either protocol implementations or end-users and their interaction with TLS warnings. As future work, we plan to conduct longitudinal studies in business environments to research long-term maintenance of keys and certificates. Furthermore, we plan to perform a participatory design workshop to determine key components within the configuration process that should (or should not) be configurable by system administrators. Based on these findings we plan to propose a new tool to make TLS easier to configure and to maintain.

List of Figures

List of Tables

Bibliography

[1] Anatomy of an attack. available online: http://blogs.rsa.com/anatomy-of-an-attack/, last accessed on 2013-07-17.

[2] Google Glass targeted as symbol by anti-tech crowd. http://edition.cnn.com/2014/04/14/tech/mobile/google-glass-attack/, accessed 10/7/2014.

[3] Google hack attack was ultra sophisticated. available online: http://www.wired.com/threatlevel/2010/01/operation-aurora/, last accessed on 2013-07-17.

[4] Google Play. https://play.google.com/store, accessed 02/09/2014.

[5] Google Safe Browsing API. https://developers.google.com/safe-browsing/, accessed 02/09/2014.

[6] Microsoft hacked: Joins apple, facebook, twitter – InformationWeek. available online: http://www.informationweek.com/security/\Attackacks/microsoft-hacked-joins-apple-facebook-tw/240149323, last accessed on 2013-07-10.

[7] Nokia Lumia Market Share. http://wmpoweruser.com/nokia-lumia-had-around-23-smartphone\-market-share-in-finland-in-q1-2013/, accessed 12/02/2014.

[8] The robin sage experiment: Fake profile fools security pros. available at http://www.networkworld.com/news/2010/070810-the-robin-sage-experiment-fake.html?t51hb, last accessed on: 2013-07-14.

[9] SSL Labs Server Rating Guide. Online at https://www.ssllabs.com/downloads/SSL_Server_Rating_Guide.pdf.

[10] Whatsapp. available online: http://www.whatsapp.com/, last accessed on 2013-07-18.

[11] The Heartbleed Bug. Online at https://heartbleed.com, 2014.

[12] D. Akhawe and A. P. Felt. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *22nd USENIX Security Symposium*, pages 257–272. USENIX Association, 2013.

[13] Alex Biryukov and Dmitry Khovratovich and Ivan Pustogarov. Deanonymisation of clients in Bitcoin P2P network. *CoRR*, abs/1405.7418, 2014.

[14] N. J. AlFardan, D. J. Bernstein, K. G. Paterson, B. Poettering, and J. C. Schuldt. On the Security of RC4 in TLS. In *22nd USENIX Security Symposium*. USENIX Association, 2013.

[15] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi. Sok: The evolution of sybil defense via social networks. *IEEE Symposium on Security and Privacy*, 2013.

[16] Android Authority. Android Jelly Bean Face Unlock 'liveness' check easily hacked with photo editing. http://www.androidauthority.com/android-jelly-bean-face-unlock-blink-hacking-105556/, last accessed 2/10/2016.

[17] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni, et al. DROWN: Breaking TLS using SSLv2. In *25th USENIX Security Symposium*. USENIX Association, 2016.

[18] Aviv, Adam J and Gibson, Katherine and Mossop, Evan and Blaze, Matt and Smith, Jonathan M. Smudge Attacks on Smartphone Touch Screens. *WOOT*, 10:1–7, 2010.

[19] G. Bader, A. Anjomshoaa, and A. Tjoa. Privacy aspects of mashup architecture. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 1141–1146, 2010.

[20] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 96–103, New York, NY, USA, 1998. ACM.

[21] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In *Recent Advances in Intrusion Detection*, pages 422–441. Springer, 2010.

[22] R. Ballagas, M. Rohs, J. G. Sheridan, and J. Borchers. Byod: Bring your own device. In *In Proceedings of the Workshop on Ubiquitous Display Environments, Ubicomp*, 2004.

[23] M. S. Barhm, N. Qwasmi, F. Z. Qureshi, and K. El-Khatib. Negotiating privacy preferences in video surveillance systems. In *Modern Approaches in Applied Intelligence*, pages 511–521. Springer, 2011.

[24] A. W. Baur, J. Bühler, M. Bick, and C. S. Bonorden. Cryptocurrencies as a disruption? empirical findings on user adoption and future potential of bitcoin and co. In *Open and Big Data Management and Innovation*, pages 63–80. Springer, 2015.

[25] Bianchi, Andrea and Oakley, Ian and Kostakos, Vassilis and Kwon, Dong Soo. The phone lock: audio and haptic shoulder-surfing resistant PIN entry methods for mobile devices. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 197–200. ACM, 2011.

[26] Bianchi, Andrea and Oakley, Ian and Kwon, Dong Soo. Spinlock: a single-cue haptic and audio PIN input technique for authentication. In *Haptic and Audio Interaction Design*, pages 81–90. Springer, 2011.

[27] Bianchi, Andrea and Oakley, Ian and Kwon, Dong Soo. Counting clicks and beeps: Exploring numerosity based haptic and audio PIN entry. *Interacting with computers*, 24(5):409–422, 2012.

[28] Bianchi, Andrea and Oakley, Ian and Lee, Jong Keun and Kwon, Dong Soo and Kostakos, Vassilis. Haptics for tangible interaction: a vibro-tactile prototype. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 283–284. ACM, 2011.

[29] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.

[30] A. Biryukov and I. Pustogarov. Bitcoin over tor isn't a good idea. *arXiv preprint arXiv:1410.6079*, 2014.

[31] Bitcoin community. Bitcoin developer guide, Oct. 2014. Accessed: 2014-10-14.

[32] Bitcoin community. Bitcoin protocol specification, Oct. 2014. Accessed: 2014-10-14.

[33] Blockchain.info. Bitcoin currency statistics, Apr. 2014. Accessed: 2014-04-05.

[34] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567. IEEE, 2012.

[35] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. 2015.

[36] J. Bonneau, S. Preibusch, and R. Anderson. A birthday present every eleven wallets? the security of customer-chosen banking pins. In *Financial Cryptography and Data Security*, pages 25–40. Springer, 2012.

[37] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 93–102. ACM, 2011.

[38] A. Bragdon, E. Nelson, Y. Li, and K. Hinckley. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 403–412. ACM, 2011.

[39] W. Breyha, D. Durvaux, T. Dussa, L. A. Kaplan, F. Mendel, C. Mock, M. Koschuch, A. Kriegisch, U. Pöschl, R. Sabet, et al. Applied Crypto Hardening. Online at `https://bettercrypto.org`, 2015.

[40] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders. Social networks and context-aware spam. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, pages 403–412, New York, NY, USA, 2008. ACM.

[41] Buschek, Daniel and De Luca, Alexander and Alt, Florian. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1393–1402. ACM, 2015.

[42] Chaos Computer Club. Chaos Computer Club breaks Apple TouchID. http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid, last accessed 11/11/2015.

[43] Cherapau, Ivan and Muslukhov, Ildar and Asanka, Nalin and Beznosov, Konstantin. On the impact of touch id on iphone passcodes. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 257–276, 2015.

[44] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner. Analyzing inter-application communication in android. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys '11, pages 239–252, New York, NY, USA, 2011. ACM.

[45] S. Chowdhury, R. Poet, and L. Mackenzie. Passhint: Memorable and secure authentication. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2917–2926, New York, NY, USA, 2014. ACM.

[46] R. Cialdini. *Influence: science and practice*. Allyn and Bacon, 2001.

[47] J. Clark and P. C. van Oorschot. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In *Symposium on Security and Privacy*, pages 511–525. IEEE, 2013.

[48] A. Dabrowski, K. Krombholz, J. Ullrich, and E. Weippl. Qr inception: Barcode-in-barcode attacks. In *Proceedings of the 4th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM 2014)*. ACM, 11 2014.

[49] A. Dabrowski, E. R. Weippl, and I. Echizen. Framework based on privacy policy hiding for preventing unauthorized face image processing. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 455–461. IEEE, 2013.

[50] De Luca, Alexander and Hang, Alina and von Zezschwitz, Emanuel and Hussmann, Heinrich. I feel like i'm taking selfies all day!: Towards understanding biometric authentication on smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI*, volume 15, pages 1411–1414, 2015.

[51] De Luca, Alexander and Harbach, Marian and von Zezschwitz, Emanuel and Maurer, Max-Emanuel and Slawik, Bernhard Ewald and Hussmann, Heinrich and Smith, Matthew. Now you see me, now you don't: protecting smartphone authentication from shoulder surfers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2937–2946. ACM, 2014.

[52] De Luca, Alexander and Lindqvist, Janne. Is Secure and Usable Smartphone Authentication Asking Too Much? *Computer*, 48(5):64–68, 2015.

[53] De Luca, Alexander and Von Zezschwitz, Emanuel and Nguyen, Ngo Dieu Huong and Maurer, Max-Emanuel and Rubegni, Elisa and Scipioni, Marcello Paolo and Langheinrich, Marc. Back-of-device authentication on smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2389–2398. ACM, 2013.

[54] T. Denning, Z. Dehlawi, and T. Kohno. In situ with bystanders of augmented reality glasses: perspectives on recording and privacy-mediating technologies. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2377–2386. ACM, 2014.

[55] Denso Wave. The QR Code Standard. `http://www.qrcode.com/en/about/standards.html`, accessed 12/02/2014.

[56] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878, 6176.

[57] P. F. Drucker. *Landmarks of tomorrow: a report on the new "post-modern" world.* Harper, New York, 1st edition, 1959.

[58] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. Analysis of the HTTPS Certificate Ecosystem. In *Internet Measurement Conference*, pages 291–304. ACM, Oct. 2013.

[59] P. Eckersley and J. Burns. An Observatory for the SSLiverse. DEF CON 18 https://www.eff.org/files/defconssliverse.pdf, July 2010.

[60] P. Eckersley and J. Burns. Is the SSLiverse a Safe Place? Chaos Communication Congress https://www.eff.org/files/ccc2010.pdf, Dec. 2010.

[61] S. Eskandari, D. Barrera, E. Stobert, and J. Clark. A first look at the usability of bitcoin key management. In *Workshop on Usable Security (USEC)*, 2015.

[62] C. Evans, C. Palmer, and R. Sleevi. Public key pinning extension for HTTP (HPKP). RFC 7469, 2015.

[63] M. H. F. Reid. An analysis of anonymity in the bitcoin system. In *2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing*, 2011.

[64] H. Fallmann, G. Wondracek, and C. Platzer. Covertly probing underground economy marketplaces. In *Seventh Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA)*, 2010.

[65] A. P. Felt, A. Ainslie, R. W. Reeder, S. Consolvo, S. Thyagaraja, A. Bettes, H. Harris, and J. Grimes. Improving SSL warnings: comprehension and adherence. In *Conference on Human Factors in Computing Systems*, pages 2893–2902. ACM, 2015.

[66] A. P. Felt, R. W. Reeder, H. Almuhimedi, and S. Consolvo. Experimenting at Scale with Google Chrome's SSL Warning. In *Conference on Human Factors in Computing Systems*, pages 2667–2670. ACM, 2014.

[67] J. Franklin, V. Paxson, S. Savage, and A. Perrig. An inquiry into the nature and causes of the wealth of internet miscreants. In *ACM Conference on Computer and Communications Security (CCS)*. ACM, November 2007.

[68] J. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology-EUROCRYPT 2015*, pages 281–310. Springer, 2015.

[69] S. L. Garfinkel, D. Margrave, J. I. Schiller, E. Nordlander, and R. C. Miller. How to make secure email easier to use. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 701–710. ACM, 2005.

[70] S. L. Garfinkel and R. C. Miller. Johnny 2: a user test of key continuity management with s/mime and outlook express. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 13–24. ACM, 2005.

[71] S. Gaw, E. W. Felten, and P. Fernandez-Kelly. Secrecy, flagging, and paranoia: adoption criteria in encrypted email. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 591–600. ACM, 2006.

[72] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun. Tampering with the delivery of blocks and transactions in bitcoin. Technical report, Cryptology ePrint Archive, Report 2015/578, 2015. http://eprint. iacr. org.

[73] S. Goldfeder, R. Gennaro, H. Kalodner, J. Bonneau, J. Kroll, E. W. Felten, and A. Narayanan. Securing Bitcoin wallets via a new DSA/ECDSA threshold signature scheme. Accessed 2015-06-09.

[74] S. Granger. Social Engineering Fundamentals, Part I: Hacker Tactics. *SecurityFocus*, 2001.

[75] N. Gruschka and M. Jensen. Attack surfaces: A taxonomy for attacks on cloud services. In *IEEE CLOUD*, pages 276–279, 2010.

[76] J. A. Halderman, B. Waters, and E. W. Felten. Privacy management for portable recording devices. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 16–24. ACM, 2004.

[77] M. Harbach, A. De Luca, and S. Egelman. The Anatomy of Smartphone Unlocking. In *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems, CHI*. 2016.

[78] M. Harbach, S. Fahl, P. Yakovleva, and M. Smith. Sorry, i don't get it: An analysis of warning message texts. In *Financial Cryptography and Data Security*, pages 94–111. Springer, 2013.

[79] Harbach, Marian and von Zezschwitz, Emanuel and Fichtner, Andreas and De Luca, Alexander and Smith, Matthew. It's a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *Symposium on Usable Privacy and Security (SOUPS)*, 2014.

[80] Z. Harris. Distributional structure. *Word 10*, 2(3), 1954.

[81] A. Harvey. CV Dazzle, 2010-2012. `http://ahprojects.com/projects/cv-dazzle`, accessed May 2nd 2013.

[82] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 129–144, Washington, D.C., Aug. 2015. USENIX Association.

[83] B. Henne, C. Szongott, and M. Smith. Snapme if you can: privacy threats of other peoples' geo-tagged media and what we can do about it. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 95–106. ACM, 2013.

[84] C. Herley. Why do nigerian scammers say they are from nigeria? In *WEIS*, 2012.

[85] C. Herley and D. Florencio. Phishing as a Tragedy of the Commons. *NSPW 2008, Lake Tahoe, CA*, 2008.

[86] C. Herley and D. Florencio. Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. Technical report, Microsoft Research, 2009.

[87] J. Hodges, C. Jackson, and A. Barth. RFC 6797: HTTP Strict Transport Security (HSTS), 2012.

[88] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar. TLS in the wild: an Internet-wide analysis of TLS-based protocols for electronic communication. In *Network and Distributed System Security Symposium*. Internet Society, Feb. 2016.

[89] R. Holz, L. Braun, N. Kammenhuber, and G. Carle. The ssl landscape: a thorough analysis of the x. 509 pki using active and passive measurements. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 427–444. ACM, 2011.

[90] T. Holz, M. Engelberth, and F. Freiling. Learning more about the underground economy: A case-study of keyloggers and dropzones. *Computer Security–ESORICS 2009*, pages 1–18, 2009.

[91] R. Hoyle, R. Templeman, D. Anthony, D. Crandall, and A. Kapadia. Sensitive lifelogs: A privacy analysis of photos from wearable cameras. In *Proceedings of The ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*, pages 1645–1648, Apr. 2015.

[92] L.-S. Huang, S. Adhikarla, D. Boneh, and C. Jackson. An Experimental Study of TLS Forward Secrecy Deployments. *Internet Computing, IEEE*, 18(6):43–51, 2014.

[93] M. Huber, S. Kowalski, M. Nohlberg, and S. Tjoa. Towards automating social engineering using social networking sites. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 3, pages 117–124. IEEE, 2009.

[94] M. Huber, M. Mulazzani, M. Leithner, S. Schrittwieser, G. Wondracek, and E. Weippl. Social snapshots: digital forensics for online social networks. In *Proceedings of the 27th Annual Computer Security Applications Conference*, 2011.

[95] M. Huber, M. Mulazzani, S. Schrittwieser, and E. Weippl. Cheap and automated socio-technical attacks based on social networking sites. In *3rd Workshop on Artificial Intelligence and Security (AISec'10)*, 10 2010.

[96] M. Huber, M. Mulazzani, E. Weippl, G. Kitzler, and S. Goluch. Friend-in-the-middle attacks: Exploiting social networking sites for spam. *IEEE Internet Computing*, 15(3):28–34, 2011.

[97] A. Hudic, K. Krombholz, T. Otterbein, C. Platzer, and E. Weippl. Automated analysis of underground marketplaces. In *IFIP International Conference on Digital Forensics*, pages 31–42. Springer Berlin Heidelberg, 2014.

[98] Huh, Jun Ho and Kim, Hyoungshick and Bobba, Rakesh B and Bashir, Masooda N and Beznosov, Konstantin. On the Memorability of System-generated PINs: Can Chunking Help? In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 197–209, 2015.

[99] G. Iachello and J. Hong. End-user privacy in human-computer interaction. *Foundations and Trends in Human-Computer Interaction*, 1(1):1–137, 2007.

[100] D. Irani, M. Balduzzi, D. Balzarotti, E. Kirda, and C. Pu. Reverse social engineering attacks in online social networks. *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 55–74, 2011.

[101] K. Ivaturi and L. Janczewski. A taxonomy for social engineering attacks. 2011.

[102] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[103] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142, Berlin, 1998. Springer.

[104] R. Kang, L. Dabbish, N. Fruchter, and S. Kiesler. "My Data Just Goes Everywhere:" User Mental Models of the Internet and Implications for Privacy and Security. In *Symposium On Usable Privacy and Security*, pages 39–52. USENIX Association, July 2015.

[105] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 523–537, 2012.

[106] Khan, Hassan and Atwater, Aaron and Hengartner, Urs. A comparative evaluation of implicit authentication schemes. In *Research in Attacks, Intrusions and Defenses*, pages 255–275. Springer, 2014.

[107] P. Kieseberg, M. Leithner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, and E. Weippl. Qr code security. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, pages 430–435. ACM, 2010.

[108] M. Korayem, R. Templeman, D. Chen, D. Crandall, and A. Kapadia. Screenavoider: Protecting computer screens from ubiquitous cameras. *arXiv preprint arXiv:1412.0008*, 2014.

[109] M. Kranch and J. Bonneau. Upgrading HTTPS in Mid-Air: An Empirical Study of Strict Transport Security and Key Pinning. In *Network and Distributed System Security Symposium*. Internet Society, Feb. 2015.

[110] K. Krombholz, A. Dabrowski, M. Smith, and E. Weippl. Ok glass, leave me alone: Towards a systematization of privacy enhancing technologies for wearable computing. In *Financial Crypto 2015 WEARABLE S&P Workshop*, 2015.

[111] K. Krombholz, A. Dabrowski, M. Smith, and E. Weippl. Exploring design directions for wearable privacy. *Under submission at the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM 2016)*, 2016.

[112] K. Krombholz, P. Frühwirt, P. Kieseberg, I. Kapsalis, M. Huber, and E. Weippl. Qr code security: A survey of attacks and challenges for usable security. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 79–90. Springer International Publishing, 2014.

[113] K. Krombholz, P. Fruhwirt, T. Rieder, I. Kapsalis, J. Ullrich, and E. Weippl. Qr code security - how secure and usable apps can protect users against malicious qr codes. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 230–237. IEEE, 2015.

[114] K. Krombholz, H. Hobel, M. Huber, and E. Weippl. Social engineering attacks on the knowledge worker. In *Proceedings of the 6th International Conference on Security of Information and Networks*, pages 28–35. ACM, 2013.

[115] K. Krombholz, H. Hobel, M. Huber, and E. Weippl. Advanced social engineering attacks. *Journal of Information Security and Applications*, 22:113–122, 2014.

[116] K. Krombholz, T. Hupperich, and T. Holz. Use the force: Evaluating force-sensitive authentication for mobile devices. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, 2016.

[117] K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl. The other side of the coin: User experiences with bitcoin security and privacy. In *Financial Cryptography and Data Security 2016*. Springer, 2016.

[118] K. Krombholz, W. Mayer, M. Schmiedecker, and E. Weippl. "i have no idea what i'm doing" - on the usability of deploying https. *Under submission at the Network and Distributed System Security Symposium (NDSS 2017)*, 2017.

[119] K. Krombholz, D. Merkl, and E. Weippl. Fake identities in social media: A case study on the sustainability of the facebook business model. *JoSSR*, 4(2):175–212, 2012.

[120] Krzysztof Okupski. Bitcoin protocol specification, Oct. 2014. Accessed: 2014-10-14.

[121] R. Kuber and W. Yu. Tactile vs graphical authentication. In *Haptics: Generating and Perceiving Tangible Sensations*, pages 314–319. Springer, 2010.

[122] M. Labs. Android Malware Spreads Through QR Code. `http://blogs.mcafee.com/mcafee-labs/android-malware-spreads-through-qr-code`, accessed 02/09/2014.

[123] J. Lazar, J. H. Feng, and H. Hochheiser. *Research methods in human-computer interaction*. John Wiley & Sons, 2010.

[124] H. K. Lee, T. Malkin, and E. Nahum. Cryptographic Strength of SSL/TLS Servers: Current and Recent Practices. In *Internet Measurement Conference*, pages 83–92. ACM, Oct. 2007.

[125] R. M. Losee. Term dependence: a basis for luhn and zipf models. *J. Am. Soc. Inf. Sci. Technol.*, 52(12):1019–1025, 2001.

[126] G. F. Lyon. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[127] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Learning to detect malicious urls. *ACM Transactions on Intelligent Systems and Technology (TIST 2011)*, 2(3):30, 2011.

[128] Malek, Behzad and Orozco, Mauricio and El Saddik, Abdulmotaleb. Novel shoulder-surfing resistant haptic-based graphical password. In *Proc. EuroHaptics*, volume 6, 2006.

[129] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[130] K. Marett, D. Biros, and M. Knode. Self-efficacy, Training Effectiveness, and Deception Detection: A Longitudinal Study of Lie Detection Training. *lecture notes in computer science*, 3073:187–200, 2004.

[131] MarketingCharts. QR Code Scanners Likely Male, Young. `http://www.marketingcharts.com/direct/qr-code-scanners-likely-male-young-21019/`, accessed 12/02/2014.

[132] W. Mayer, A. Zauner, M. Schmiedecker, and M. Huber. No Need for Black Chambers: Testing TLS in the E-mail Ecosystem at Large. *arXiv preprint arXiv:1510.08646*, Oct. 2015.

[133] P. Mcnamee and J. Mayfield. Character n-gram tokenization for european language text retrieval. *Inf. Retr.*, 7(1-2):73–97, Jan. 2004.

[134] R. Meunier. Pattern languages of program design. chapter The pipes and filters architecture, pages 427–440. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995.

[135] K. Miller, J. Voas, and G. Hurlburt. Byod: Security and privacy considerations. *IT Professional*, 14(5):53–55, 2012.

[136] K. Mitnick and W. Simon. *The Art of Deception: Controlling the Human Element of Security*. Wiley, 2002.

[137] F. Mohd Foozy, R. Ahmad, M. Abdollah, R. Yusof, and M. Mas' ud. Generic taxonomy of social engineering attack. 2011.

[138] B. Möller, T. Duong, and K. Kotowicz. This POODLE bites: exploiting the SSL 3.0 fallback. *Google, Sep*, 2014.

[139] T. Moore and N. Christin. Beware the middleman: Empirical analysis of bitcoin-exchange risk. In *Financial Cryptography and Data Security*, pages 25–33. Springer, 2013.

[140] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl. Dark clouds on the horizon: using cloud storage as attack vector and online slack space. In *Proceedings of the 20th USENIX conference on Security*, SEC'11, pages 5–5, Berkeley, CA, USA, 2011. USENIX Association.

[141] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. online, Dec 2008.

[142] R. Nelson. Methods of Hacking: Social Engineering. online, 2008. available at: http://www.isr.umd.edu/gemstone/infosec/ver2/ papers/socialeng.html, last accessed on 2013-07-04.

[143] NIST. FIPS 180-4: Secure Hash Standard (SHS), Mar. 2012.

[144] M. Oltrogge, Y. Acar, S. Dechand, M. Smith, and S. Fahl. To Pin or Not to Pin—Helping App Developers Bullet Proof Their TLS Connections. In *24th USENIX Security Symposium*, pages 239–254. USENIX Association, Aug. 2015.

[145] F. Pallas, M.-R. Ulbricht, L. Jaume-Palasí, and U. Höppner. Offlinetags: A novel privacy approach to online photo sharing. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 2179–2184, New York, NY, USA, 2014. ACM.

[146] K. Parsons, A. McCormac, M. Pattinson, M. Butavicius, and C. Jerram. Phishing for the truth: A scenario-based experiment of users' behavioural response to emails. In L. Janczewski, H. Wolfe, and S. Shenoi, editors, *Security and Privacy Protection in Information Processing Systems*, volume 405 of *IFIP Advances in Information and Communication Technology*, pages 366–378. Springer Berlin Heidelberg, 2013.

[147] S. N. Patel, J. W. Summet, and K. N. Truong. Blindspot: Creating capture-resistant spaces. In *Protecting Privacy in Video Surveillance*, pages 185–201. Springer, 2009.

[148] S. Patil, G. Norcie, A. Kapadia, and A. J. Lee. Reasons, rewards, regrets: privacy considerations in location sharing as an interactive practice. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 5. ACM, 2012.

[149] J. Payne, G. Jenkinson, F. Stajano, M. A. Sasse, and M. Spencer. Responsibility and Tangible Security: Towards a Theory of User Acceptance of Security Tokens. *arXiv preprint arXiv:1605.03478*, 2016.

[150] N. Perlroth. Chinese hackers infiltrate new york times computers, Jan. 2013. available at `https://www.nytimes.com/2013/01/31/technology/chinese-hackers-infiltrate-new-york-times-computers.html`, last accessed on: 2013-07-01.

[151] R. Potharaju, A. Newell, C. Nita-Rotaru, and X. Zhang. Plagiarizing smartphone applications: attack strategies and defense techniques. In *Proceedings of the 4th international conference on Engineering Secure Software and Systems*, ESSoS'12, pages 106–120, Berlin, Heidelberg, 2012. Springer-Verlag.

[152] S. Preibusch. Privacy behaviors after snowden. *Communications of the ACM*, 58(5):48–55, 2015.

[153] T. Qin and J. Burgoon. An Investigation of Heuristics of Human Judgment in Detecting Deception and Potential Implications in Countering Social Engineering. *Intelligence and Security Informatics, 2007 IEEE*, pages 152–159, 2007.

[154] J. Radianti, E. Rich, and J. Gonzalez. Using a mixed data collection strategy to uncover vulnerability black markets. In *Workshop for Information Security and Privacy*. Citeseer, 2007.

[155] N. Raval, A. Srivastava, K. Lebeck, L. Cox, and A. Machanavajjhala. Markit: Privacy markers for protecting visual secrets. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1289–1295. ACM, 2014.

[156] I. Ristic. Internet SSL survey 2010. *Black Hat USA*, 3, 2010.

[157] I. Ristic. State of SSL. *Talk at InfoSec World*, 2011.

[158] J. C. Roberts, II and W. Al-Hamdani. Who can you trust in the cloud? a review of security issues within cloud computing. In *Proceedings of the 2011 Information Security Curriculum Development Conference*, InfoSecCD '11, pages 15–19, New York, NY, USA, 2011. ACM.

[159] F. Roesner, D. Molnar, A. Moshchuk, T. Kohno, and H. J. Wang. World-driven access control for continuous sensing. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1169–1181. ACM, 2014.

[160] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[161] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Goldberg. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In *Protecting Privacy in Video Surveillance*, pages 65–89. Springer, 2009.

[162] S. Schrittwieser, P. Fruehwirt, P. Kieseberg, M. Leithner, M. Mulazzani, M. Huber, and E. Weippl. Guess Who Is Texting You? Evaluating the Security of Smartphone Messaging Applications. In *Network and Distributed System Security Symposium (NDSS 2012)*, 2 2012.

[163] S. Schrittwieser, M. Mulazzani, and E. Weippl. Ethics in security research - which lines should not be crossed? In *Proceedings of the 2013 Cyber-Security Research Ethics Dialog & Strategy Workshop (CREDS 2013)*, 2013.

[164] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[165] J. Seeburger. No cure for curiosity: linking physical and digital urban layers. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design (NordiCHI 2012)*, pages 247–256. ACM, 2012.

[166] V. Sharma. A Study of Malicious QR Codes. *International Journal of Computational Intelligence and Information Security*, 3(5), 2012.

[167] Y. Sheffel, R. Holz, and P. Saint-Andre. Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS(DTLS). RFC 7457 (Proposed Standard), 2015.

[168] Y. Sheffer, R. Holz, and P. Saint-Andre. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). RFC 7525 (Proposed Standard), 2015.

[169] S. Sheng, L. Broderick, C. A. Koranda, and J. J. Hyland. Why johnny still can't encrypt: evaluating the usability of email encryption software. In *Symposium On Usable Privacy and Security*, 2006.

[170] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM, 1996.

[171] SocialEngineer. What is phishing - paypal phishing examples. available online: http://www.social-engineer.org/wiki/archives/Phishing/Phishing-PayPal.html, last accessed on 2013-07-04.

[172] Song, Youngbae and Cho, Geumhwan and Oh, Seongyeol and Kim, Hyoungshick and Huh, Jun Ho. On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2343–2352. ACM, 2015.

[173] Sophos. Sophos facebook id probe shows 41% of users happy to reveal all to potential identity thieves, 2007. available online: http://www.sophos.com/en-us/press-office/press-releases/2007/08/facebook.aspx, last accessed on 2013-07-13.

[174] S. Stasiukonis. Social Engineering, the USB Way. 2006. available at http://www.darkreading.com/security/perimeter/show Article.jhtml?articleID=208803634, last accessed on: 2013-07-02.

[175] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, SNS '11, pages 8:1–8:8, New York, NY, USA, 2011. ACM.

[176] E. Stobert and R. Biddle. The Password Life Cycle: User Behaviour in Managing Passwords. In *Symposium On Usable Privacy and Security*, pages 243–255. USENIX Association, July 2014.

[177] A. Strauss, J. Corbin, et al. *Basics of qualitative research*, volume 15. Newbury Park, CA: Sage, 1990.

[178] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of ssl warning effectiveness. In *USENIX Security Symposium (SEC 2009)*, pages 399–416, 2009.

[179] Symantec. Symantec report on the underground economy july 07–june 08. Technical report, Symantec, 2008.

[180] L. Tam, M. Glassman, and M. Vandenwauver. The psychology of password management: a tradeoff between security and convenience. *Behav. Inf. Technol.*, 29(3):233–244, May 2010.

[181] R. Templeman, M. Korayem, D. J. Crandall, and A. Kapadia. Placeavoider: Steering first-person cameras away from sensitive spaces. In *NDSS*, 2014.

[182] theregister.co.uk. That square QR barcode on the poster? Check it's not a sticker. `http://www.theregister.co.uk/2012/12/10/qr_code_sticker_scam/`, accessed 11/31/2013.

[183] R. Thomas and J. Martin. The underground economy: Priceless. In *USENIX ; LOGIN:*, 2006.

[184] H. Thompson. The human element of information security. *Security Privacy, IEEE*, 11(1):32–35, 2013.

[185] TrustedSec. Social-engineer toolkit, 2013. available online at: https://www.trustedsec.com/downloads/social-engineer-toolkit/, last accessed 03/12/2013.

[186] T. Vidas, E. Owusu, S. Wang, C. Zeng, L. F. Cranor, and N. Christin. QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks. In *Proceedings of the 2013 Workshop on Usable Security (USEC 2013)*, 2013.

[187] von Zezschwitz, Emanuel and De Luca, Alexander and Brunkow, Bruno and Hussmann, Heinrich. SwiPIN: Fast and secure pin-entry on smartphones. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI*, volume 15, pages 1403–1406, 2015.

[188] S. Weber, M. Harbach, and M. Smith. Participatory Design for Security-Related User Interfaces. In *USEC*. Internet Society, Feb. 2015.

[189] C. Wharton, J. Rieman, C. Lewis, and P. Polson. The cognitive walkthrough method: A practitioner's guide. In *Usability inspection methods*, pages 105–140. John Wiley & Sons, Inc., 1994.

[190] A. Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Usenix Security*, volume 1999, 1999.

[191] I. H. Witten and E. Frank. *Data mining : practical machine learning tools and techniques*. Elsevier, Morgan Kaufman, Amsterdam [u.a.], 2. ed. edition, 2005.

[192] Xyologic. XYO - Apps to the people. http://xyo.net, accessed 12/09/2013.

[193] T. Yamada, S. Gohshi, and I. Echizen. Use of invisible noise signals to prevent privacy invasion through face recognition from camera images. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 1315–1316, New York, NY, USA, 2012. ACM.

[194] T. Yamada, S. Gohshi, and I. Echizen. Privacy Visor: Method for Preventing Face Image Detection by Using Differences in Human and Device Sensitivity. 2013. unpublished, under review for CMS 2013.

[195] H. Yao and D. Shin. Towards Preventing QR Code Based Attacks on Android Phone Using Security Warnings. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIA CCS'13)*, pages 341–346, 2013.

[196] R. Yus, P. Pappachan, P. K. Das, E. Mena, A. Joshi, and T. Finin. Demo: Faceblock: privacy-aware pictures for google glass. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 366–366. ACM, 2014.

[197] J. Zhuge, T. Holz, C. Song, J. Guo, X. Han, and W. Zou. Studying malicious websites and the underground economy on the chinese web. Technical report, 2007.

Appendix

Wearable Privacy

Interview Questions

1. Do you know what this is?
2. Did you know that you can record video with those kinds of glasses?
3. How do you feel about being around someone who is wearing those kinds of glasses?
4. Do you have any privacy concerns?
5. On a scale from 1-5, how much would you be interested in a technology or product to protect your privacy?
6. Would you want someone to ask for permission before recording a video?
7. Would you want to be asked for permission before being recorded?
8. Which of the proposed methods would you prefer? 5 point Likert scale?
9. Why would you prefer this method?
10. Imagine you are in a cafe/at the beach instead of in this cafe/at this beach, which method would you prefer and why?
11. Back at the cafe/beach, would you still prefer this method?
12. How much would you pay for the presented techniques to express your privacy preference?
13. Would you buy additional clothing or accessories such as bikinis, t-shirts, scarves?
14. How much would you pay for such an app?
15. How much would you pay for such an electronic device?

Force-PINs

Lab Study Questionnaire

The following questions were answered by the participants of the lab study after they used the three different types of PINs in a randomized order (four-digit/six-digit/force-PIN).

Demographics

1. What was your ID during the lab experiments?
2. Gender
3. Age
4. Are you studying IT security or are you working in an IT security-related field? *(yes/no)*
5. What kind of smartphone are you currently using? *(single-choice: iPhone, Android, Windows Phone, Other, I don't use a smartphone)*
6. What methods are you currently using to unlock your smartphone? *(multiple-choice: 4-digit PINs, 6-digit PINs, character and digit password, unlock pattern, fingerprint sensor, Android Smartlock, none)*

Estimated security and usability of the three PIN types

1. Which of the three PIN methods do you think is the most secure? *(single-choice: 4-digit PINs, 6-digit PINs, force-PINs, I don't know)*
2. Which of the three PIN methods do you think is the easiest to remember? *(single-choice: 4-digit PINs, 6-digit PINs, force-PINs, I don't know)*
3. Which of the three PIN methods do you think is the least secure? *(single-choice: 4-digit PINs, 6-digit PINs, force-PINs, I don't know)*
4. Which of the three PIN methods do you think is the most time-consuming? *(single-choice: 4-digit PINs, 6-digit PINs, force-PINs, I don't know)*
5. Which of the three PIN methods do you think is the hardest to remember? *(single-choice: 4-digit PINs, 6-digit PINs, force-PINs, I don't know)*
6. Which of the three PIN methods do you think is the least time-consuming? *(single-choice: 4-digit PINs, 6-digit PINs, force-PINs, I don't know)*

Open-ended questions

1. What did you like about force-PINs?
2. What did you NOT like about force-PINs?
3. Can you think of a situation where force-PINs would be particularly useful?

Field Study Debriefing Interviews

1. Where did you use force-PINs?
2. What did you like about force-PINs?
3. What did you NOT like about force-PINs?
4. Can you think of a situation where force-PINs were particularly useful?
5. Can you think of a situation where force-PINs were annoying?
6. Is there anything else you would like to let us know?

Study Apps

The following screenshots show the user interface of the apps used for the lab and field study. Figure 1a and Figure 1b were used to evaluate force-PINs in the lab study. The apps for the other two conditions had the same layout but evaluated four-digit and six-digit PINs, respectively. Figure 1c shows the main screen of the app used in the field study.

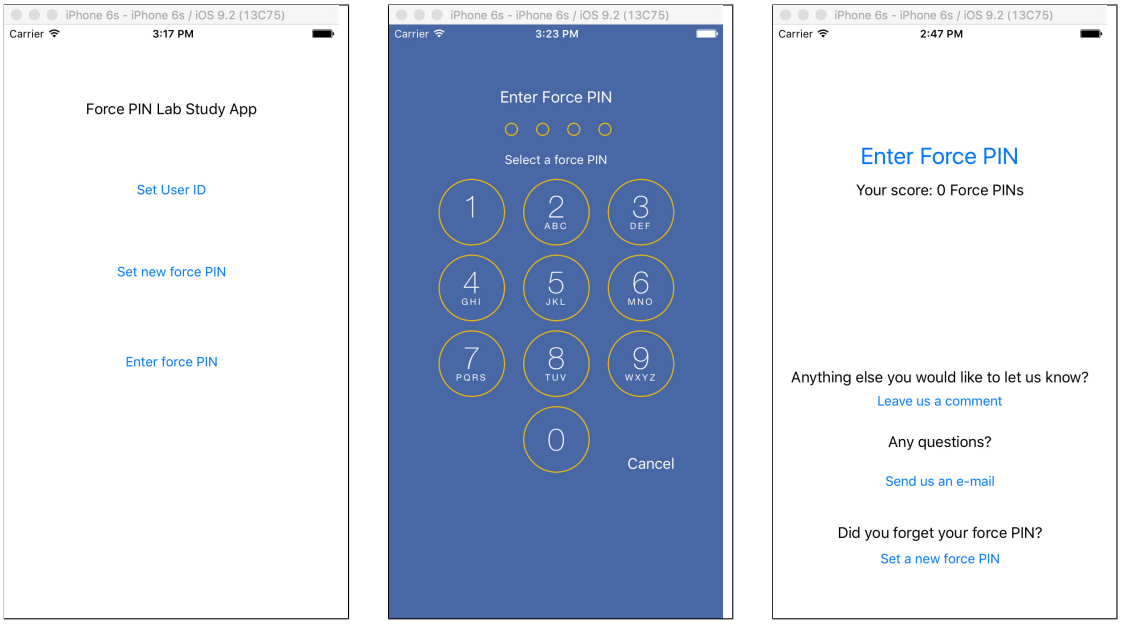
Bitcoin

Interview Questions

Questions with answer options as "( )" are multiple choice checkboxes whereas answer possibilities marked alphabetical e.g. "a)" are single selections.

BTC Demographics

Q1 Please input which year you started using Bitcoin: (2009; 2010; 2011; 2012; 2013; 2014; 2015)



(a) Main screen of the lab study app.

(b) Lock screen of both the lab study and field study app.

(c) Main screen of the field study app.

Figure 1: Screenshots of the study force-PIN apps.

Q2 Select which main features are responsible for you using Bitcoin (multiple selections possible): the opportunity of financial gain; Curiosity; Anonymous nature, Decentralized nature; A friend/colleague suggested to me to start using Bitcoin; The possibility to internationally transfer money with relatively low fee; The possibility to accept bitcoins for my services or for my products; Other

Q3 What is the estimated sum of bitcoins you are holding? I hold approximately VALUE; I do not want to specify

Q4 Please provide what services or products you pay for with bitcoins (multiple selections possible): Bars, restaurants; Bitcoin gift cards; Donations, tipping; Drugs; Gambling sites; Hotels, travel; Online marketplaces and auctions; Online shopping (Newegg, ...); Altcoin (e.g. Litecoin, ...); Physical stores that accept bitcoins; Underground marketplaces; Virtual goods (webhosting, online newspapers, ...); Medium for currency exchange; Other

Q5 What do you think are the most likely risks associated with Bitcoin?

Q6 Please select the crypto currencies you are holding or using besides Bitcoin (multiple selections possible): I do not use other crypto currencies; BanxShares; BitShares; BlackCoin; Bytecoin; Counterparty; Dash; Dogecoin; Litecoin; MaidSafeCoin;

MonaCoin; Monero; Namecoin; Nxt; Peercoin; Primecoin; Ripple; Startcoin; Stellar; SuperNET; Vertcoin; YbCoin; Other

Q7 Select the Bitcoin exchanges you have used in the past or you are using on regularly (multiple selections possible): None; BanxIO; Bitcoin Exchange Thailand; Bittrex; Bitcoin Indonesia; bitcoin.de; Bitfinex; Bitstamp; BitX South Africa; BTC-e; BTC38; BTCChina; CCEDK; Cryptsy; Gatecoin; hibtc; Kraken; Mt. Gox; OKCoin; Poloniex; QuadrigaCX; The Rock Trading; VirWox; Other:

Q8 What do you think are the greatest benefits of Bitcoin?

Q9 How often do you perform Bitcoin transactions? At least once a day; At least once a week; At least once a month; At least once every six months; At least once a year; Less than once a year

BTC Wallets

Q10 Please tick which wallets you are currently using (multiple selections possible): Airbitz; Armory; Bitcoin Core; Bitcoin Wallet (Schildbach Wallet); BitGo; Bither; breadwallet; Circle; Coinapult; Coinbase; Coinkite; Coinomi; Electrum; Green Address; Hive; Ledger Nano; mSIGNA; MultiBit; Mycelium; Ninki; TREZOR; Xapo; Not in list

Q11 Why did you choose to use multiple wallets to manage your bitcoins?

Wallet Usage

For every selected wallet in Q10 we asked the following questions.

Q12 Why did you choose ⟨wallet-name⟩ to manage your Bitcoins?

Q13 How many bitcoins do you have approximately in this wallet? I hold approximately ⟨textfield⟩ bitcoins; I do not want to specify

Q14 Is this wallet password protected? Yes; No; I do not care; I do not know

Q15 Is this wallet encrypted? Yes; No; I do not care; I do not know

Q16 Is this wallet backed up? Yes; No; I do not know

BTC Mining

Q17 Are you currently mining bitcoins? Yes, since VALUE ; No, but I have mined from VALUE to VALUE; No, I have never mined bitcoins

Q18 How many bitcoins have you mined in total? I mined approximately VALUE; I do not want to specify

Q19 Do you or have you participated in mining pools? Yes; No

Q20 Please tick the names of the mining pools you have or are participating in (multiple mentions possible): 21 Inc.; AntPool; Bitcoin Affiliate Network; BitFury; BitMinter; Bitsolo; BTCChina Pool; BTC Guild; BTC Nuggets; BW.COM; EclipseMC; Eligius; F2Pool; GHash.IO; Kano CKPool; KnCMiner; MegaBigPower; P2Pool; Slush; Telco 214; Other

BTC Server

Q21 Do you run a full Bitcoin server that is reachable for others from the Internet? Yes; No

Q22 Please provide some reasons on why you operate a full Bitcoin server (multiple selections possible): Fast transaction propagation; Double-spending detection; Network analysis; Support the Bitcoin network; Other

BTC Security Risks

(all on a 7 Point Likert-Scale from "High" to "Low")

Q23 How would you estimate the risk of monetary loss for Bitcoin compared to credit cards?

Q24 How high do you think is the risk of becoming a victim of a successful double spending attack?

Q25 How high or low would you estimate the risk for malware that steals your Bitcoins?

Q26 How would you estimate the risk of monetary theft in case the device with your wallet gets lost or stolen?

Q27 How would you estimate the risk of de-anonymization?

Q28 How high do you think the risk of cryptographic flaws is?

Q29 How high do you think is the risk of security vulnerabilities in hosted/web wallets or Exchange services?

Q30 How high do you think is the risk of key loss due to a device failure?

Q31 How high do you think is the risk that the Bitcoin network is temporarily not available?

Q32 How high do you think is the risk of a centralization of mining?

Q33 How high do you think is the risk of a strong fluctuation in the Bitcoin exchange rate (e.g. BTC to USD and vice versa)?

BTC Anonymity

Q34 Do you think that Bitcoin usage is anonymous? Yes, Bitcoin is fully anonymous; No, Bitcoin is not anonymous; Not per se, but it can be used in an annonymouse manner

Q35 Do you think it is possible to follow your transactions? Yes; No

Q36 Have you ever used Bitcoin over Tor (Tool tip with description of Tor)? Yes; No

Q37 Do you take additional steps to ensure your privacy using Bitcoin?Yes; No

BTC Security Breaches

Q38 Have you ever lost your bitcoins or Bitcoin keys? Yes; No

Q39 Please select the reason for your key/Bitcoin loss (multiple selections possible): Hardware failure (e.g. hard drive broke, etc); Software failure (e.g. keyfile corruption, etc); Self induced event (e.g. hard drive formatted, physical device lost, etc); Malicious event (e.g. malware, hacker, etc); Other

Q40 Have you been able to recover your keys? Yes; No + free text

Q41 How many bitcoins did you loose due to this incident? VALUE bitcoins; I do not want to specify

Q42 Please select the security incidents you have been affected by (multiple selections possible): None; Mt. Gox incident; Silk Road bust; inputs.io hack; Pony botnet malware; Pyramid schemes / HYIPS (High yield investment programs); Mining hardware scams (Labcoin, Active Mining Corporation, Ice Drill, AsicMiningEquipment.com, Dragon-Miner.com, ...); Mining pool scams; Scam wallets; Bitcoin exchange scam; Other

Q43 How did you deal with the incident?

Q44 What was the approximate value of your lost bitcoins in USD?VALUE USD; I do not want to specify; I do not know

Demographics

Q45 Please provide your age:

Q45 Please provide your gender: Female; Male; Do not want to specify

Q46 Please select your highest completed level of education: Did Not Complete High School; High School/GED; Some College; Bachelor's Degree; Master's Degree; Advanced Graduate work or Ph.D.; Not Sure

Q47 Do you work or study in a computer science related field? Yes; No

Q48 How would you describe yourself in terms of privacy behaviour? Continouos slider between "I am not concerned about my privacy" and "I would describe myself as a privacy fundamentalist"

End

Q49 You can enter your Bitcoin address in the textfield below. Please make sure that your address is correct in order to receive your incentive.

Q49 This is the place where you can provide suggestions, complaints or any other information we may have forgotten to ask in the questionnaire.

Address Signature

```
./bitcoin-cli signmessage 12yeU5ymM67SL5UWVSwErAgwVwwaTd1Nma \
"https://www.soscisurvey.de/BTC_study/"
HzzNxFmeRhbhAwVZ4DsraBkXkW7JYjO0tAlIPAnHB2z5P12eddFilWXJmwGm\
PkgS/v8W0DNr0Z1qLwroPbWWMoE=
```

Reference link issue

We had a problem in our implementation of this last page of the survey which also showed the link to the survey containing a random reference which should identify this particular participant in our rewarding scheme. If the CAPTCHA was not solved successfully the side reloads itself and would also calculate and show a different reference link. The references link will only be stored and linked to this particular participant if the CAPTCHA is entered correctly. Therefore, all users which just copied the first link and then entered a wrong CAPTCHA distributed a link we where not able to attribute correctly at the end of the survey.

TLS User Study

Lab Study Assignment

You are the system administrator at a SME (small and medium-sized enterprise). Your company runs a web portal and your boss instructed you to secure the communication by using TLS. Unfortunately you only have a very limited amount of time because your company will also soon be under security audit. This is why you should start right away deploying TLS. Make your configuration as secure as possible.

System Configuration

- The company's web server (Apache2) is currently found at *http://web.local* on Port 80. There is only HTTP activated. No TLS configuration is made so far.
- You can connect to the web server with the command *ssh web* The username is *pi*, the password is *raspberry*. There is no root password, so you can just use *sudo* to execute commans as root user.
- You will have to use a Certificate Authority. You find a CA at *https://ca.local* Your client's Firefox trusts this CA called *TLS Userstudy Root CA*. You can test the certificate validation with this browser. The DNS names of both servers are locally configured at your client.

Post Lab Study Questionnaire

Demgraphics

- Participant ID (assigned prior to the lab experiments)
- Age
- Gender
- Months of industry experience

Experience with TLS

- Are you currently in charge of a web server? (Yes, I'm currently administrating a company web server./ Yes, I'm currently administrating a private web server./ Yes, I'm currently administrating at a profit/non-profit association. /No.)
- Have you ever installed and configured SSL/TLS before? (yes/no)
- Have you ever worked as a system administrator before? (yes/no)

Reflections on the study task

- Did you finish the TLS installation in the given time? (yes, no, I'm not sure)
- If you didn't finish the TLS installation in the given time, which steps are still missing to secure the communication? (open text)
- How difficult did you find TLS deployment? (Likert scale: very easy to very difficult)
- What did you find particularly difficult? (open text)

- What do you think are the key usability pitfalls of TLS deployment? (open text)
- What would you recommend a system administrator who has to deploy TLS? (open text)
- Is there anything else you would like to let us know? (open text)

Interview Questions - Expert Interviews

- As an auditor, how do you usually proceed to evaluate the security of a TLS configuration?
- What are the main vulnerabilities/configuration mistakes that you encounter as an auditor?
- What botheres admins/CSOs the most regarding TLS?
- What are the most critical steps in TLS deployment?
- How should the deployment process be improved?
- What piece of advice would you generally give to anyone in charge of securing communication over HTTPS?

Detailed Evaluation Criteria

Grade with Trust Issues Ignored The overall grade for the configuration with a valid certificate. The grade is calculated based on the grading scheme from [9]. The score is based on individual ratings for protocol support (30%), key exchange (30%) and cipher strength (40%). The grade is issued based on the following cumulative scores:

- A: score >= 80
- B: score >= 65
- C: score >= 50
- D: score >= 35
- E: score >= 20
- F: score < 20

Errors/Warnings/Highlights: This refers to remarks that impacted the overall grading. The detailed description of these justifications is shown in Table 8.3.

Cipher Strength Score: This is represented by a number between 0 and 100, with 100 being the best possible. The cipher strength score contributes 40% to the overall grade. As weak symmetric ciphers can be easily broken by attackers, it is essential

to the overall configuration that strong ciphers are used. SSL Labs evaluate ciphers based on an average cipher between the strongest and weakest. The scores are rated as follows:

- 0 bits (no encryption): 0
- < 128 bits (e.g., 40, 56): 20
- < 256 bits (e.g., 128, 168): 80
- >= 256: 100

Key Exchange Score: As described in [9], the key exchange phase serves two functions: (1) authentication to verify the identity of the other party and (2) safe generation and exchange of secret keys to be used for the remaining session. Also, exportable key exchanges where only a part of the key is exchanged can make the session keys easier to compromise. Key exchange without authentication is vulnerable to MITM attacks and allows an attacker to gain access to the communication channel. Furthermore, the strength of the server's private key is crucial. The stronger it is, the more difficult it is to break the key exchange phase. Some servers use the private key just for authentication and not for the key exchange mechanism. Popular algorithms are the Diffie-Hellman key exchange (DHE) and its elliptic curve version (ECDHE). As in [9], the rating is calculated as follows:

- Weak key or anonymous key exchange (e.g., Anonymous Diffie-Hellman): 0
- Key or DH parameter strength < 512 bits: 20
- Exportable key exchange limited to 512 bits: 40
- Key or DH parameter strength < 1024 bits: 40
- Key or DH parameter strength < 2048 bits: 80
- Key or DH parameter strength < 4096 bits: 90
- Key or DH parameter strength >= 4096 bits: 100

Protocol Support Score [9] Several (older) versions of TLS have known weaknesses or are vulnerable to well-known attacks. The configuration is graded as follows with respect to the activated TLS versions. Again, if multiple versions are supported, the average between the best and worst protocol score is considered.

- SSL 2.0: 0
- SSL 3.0: 80
- TLS 1.0: 90
- TLS 1.1: 95
- TLS 1.2: 100

Common Name: This refers to the common name field specified in the CSR which specifies a FQDN (and respective subdomains if applicable) the certificate is issued for.

Key Size: This refers to the size of the server's key pair.

Certificate Chain Length: This refers to the length of the certificate chain, including the server's certificate and certificates of intermediate CAs, and the certificate of a root CA trusted by all parties in the chain. Every intermediate CA in the chain holds a certificate issued by the CA one level above it in the trust hierarchy. In our example, the ideal length is 3.

Used Provided CA to sign: In order to remove the bias from different CAs with varying usability, we implemented our own CA and provided the link to this CA in the assignment. Two participants did not use this CA and generated self-signed certificates instead.

Encrypted Private Key indicates whether the server's private key was encrypted by the study participant.

SSL 2 – TLS 1.2 indicates which protocol versions are supported.

RC4 support: To date, RC4 is considered weak and should therefore not be supported, unless required for compatibility reasons as found in [14].

Vulnerable to POODLE indicates whether the configuration is vulnerable to POODLE [138].

Forward Secrecy indicates whether the configuration supports ciphers with forward secrecy (e.g., ECDHE).

HSTS indicates whether *HTTP Strict Transport Security* is configured. The security benefit of HSTS is that it forces secure communication with websites that use it by automatically converting all plain text and disabling click-through certificate warnings. If a client does not support HSTS, it simply ignores the header. Hence, activating HSTS enhances security with minimal effort without impact on compatibility.

HPKP indicates whether *Public Key Pinning* is used, which is a useful feature to prevent attacks and making the public aware of them.