5-1-2020

Bridging the Gap Between Intent and Outcome: Knowledge, Tools & Principles for Security-Minded Decision-Making

Vijay Harshed Kothari
Dartmouth College

BRIDGING THE GAP BETWEEN INTENT AND OUTCOME:
KNOWLEDGE, TOOLS & PRINCIPLES FOR SECURITY-MINDED DECISION-MAKING

Dartmouth Computer Science Technical Report TR2020-880

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Vijay Harshed Kothari

Guarini School of Graduate and Advanced Studies

Dartmouth College

Hanover, New Hampshire

May, 2020

Examining Committee:

Sean Smith, Ph.D., Chair

Sergey Bratus, Ph.D.

Venkatramanan Siva Subrahmanian, Ph.D

Jim Blythe, Ph.D.

Ross Koppel, Ph.D.

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies

Abstract

Well-intentioned decisions—even ones intended to improve aggregate security— may inadvertently jeopardize security objectives. Adopting a stringent password composition policy ostensibly yields high-entropy passwords; however, such policies often drive users to reuse or write down passwords. Replacing URLs in emails with "safe" URLs that navigate through a gatekeeper service that vets them before granting user access may reduce user exposure to malware; however, it may backfire by reducing the user's ability to parse the URL or by giving the user a false sense of security if user expectations misalign with the security checks delivered by the vetting process. A short timeout threshold may ensure the user is promptly logged out when the system detects they are away; however, if an infuriated user copes by inserting a USB stick in their computer to emulate mouse movements, then not only will the detection mechanism fail but the insertion of the USB stick may present a new attack surface. These examples highlight the disconnect between decision-maker intentions and decision outcomes. Our focus is on bridging this gap.

This thesis explores six projects bound together by the core objective of empowering people to make decisions that achieve their security and privacy objectives. First, we use grounded theory to examine Amazon reviews of password logbooks and to obtain valuable insights into users' password management beliefs, motivations, and behaviors. Second, we present a discrete-event simulation we built to assess the efficacy of password policies. Third, we explore the idea of supplementing language-theoretic security with human-computability boundaries. Fourth, we conduct an eye-tracking study to understand users' visual processes while parsing and classifying URLs. Fifth, we discuss preliminary findings from a study conducted on Amazon Mechanical Turk to examine why users fall for unsafe URLs. And sixth, we develop a logic-based representation of mismorphisms, which allows us to express the root causes of security problems. Each project demonstrates a key technique that can help in bridging the gap between intent and outcome.

Acknowledgments

I am indebted to my graduate advisor and mentor, Sean Smith, for the support, sage advice, and direction that he's provided me throughout graduate school. In many instances where I didn't know how best to tackle a problem I was facing, I attempted to emulate him. And, while I have fallen short plenty of times through faults of my own, I believe this approach has generally been the correct one. I've also finally come to recognize the wisdom of the Voltairean principle to which he often refers: don't let perfect be the enemy of the good.

I thank my committee members. I appreciate the occasionally lengthy, always interesting conversations with Sergey Bratus, who has presented crisp, well-reasoned perspectives I might not otherwise have encountered, both within the security and privacy domains and external to them; both Anna and Sergey have been incredibly kind, supportive, and educative throughout my PhD—and for this, I am immensely thankful to them both. Ross Koppel has been extremely supportive, and he has invested a great deal of time in providing me with thoughtful feedback and advice; he has also taught me about a variety of methods and techniques in sociology, such as those employed in grounded theory. Jim Blythe has consistently provided me with (good) direction, has taught me much about agent-based simulation, and has been a valuable resource throughout grad school. Venkatramanan Siva Subrahmanian (VS) has also given me valuable feedback on my work, and it is always a pleasure to speak with him.

Additionally, I would like to thank Dave Kotz who served on my RPE committee. I'd also like to thank Kimo Johnson, Prasad Jayanti, Peter Winkler, and Amit Chakrabarti, who, in addition to being great lecturers, have been welcoming and have spent abundant time educating me on topics pertaining to their academic expertise and more generally.

I am very grateful to my undergraduate advisor, Rajiv Gandhi. His patience, empathy, knowledge, passion for teaching, and commitment to providing his students with every opportunity available make him an exemplary mentor to many students over the years. And I'm glad to be one. I am also grateful to Suneeta Ramaswami for providing me the opportunity to pursue research in computational geometry. I would like to thank E. Roger Cowley, L. John Gagliardi, Aram Mekjian, and William Steiger, who have been great teachers who have also played instrumental roles in my academic education and my research trajectory. I am grateful to David Williamson and David Shmoys for providing me with an early, draft copy of their book, The Design of Approximation Algorithms, which served me well in learning about approximation algorithms and pursuing research in the field. I am also glad to have met many fellow students and all-around great people during my time at Rutgers, including, but not limited to: Josh Wetzel, James Davis, Sean Lowen, Elaine Zamora, and Garen Eisen.

It has been a joy and privilege to be in the company of fellow labbies: Jason Reeves, Rebecca Shapiro, Ira Ray Jenkins, Prashant Anantharaman, Michael Millian, Peter Johnson, John Peter Brady, Sameed Ali, Nathan Reitinger, Gabriel Weaver, Taylor Campbell, Vineetha Paruchuri, Stefan Boesen, Zephyr Lucas, Joshua Ackerman, Anmol Chachra, Chen Qin, Kirti Rathore, Samuel Tan Jie, Syed Tanveer, Chris Novak, Bruno Korbar, Ryan Amos, Patrick Flathers, Galen Brown, Ivan Antoniv, Linda Xiao, and many, many more. And, of course, it's also been great to get to know and learn from members of sister labs: Travis Peters, Shrirang Mare, Aarathi Prasad, Tim

Pierson, Taylor Hardin, Patrick Proctor, Chongyang Bai, Qian Han, Sougata Sen, and others. I'd like to thank other students and also attendees of various security-focused lunches: John Gilman, Weijia Mao, Adrian Kostrubiak, Sandeep Nuckchady, Shaun Akhtar, Steve Cannon, Wes Kendrick, Mohammad Haris Baig, Natalie Afonina, Mahesh Devalla, and numerous others.

I am also grateful for the opportunity to collaborate with my colleagues. I have mentioned many already, but I also would like to thank Niveta Ramkumar, Andrew Campbell, Theo Nkengfack, Sean Serpa, and Caitlin Mills. I have learned much from them.

It has been an absolutely pleasure to interact with members of the Institute for Security, Technology, and Society (ISTS) over the years: Bill Nisen, Julie Gilman, Karen Page, and Thomas Candon. They have all been empathetic and encouraging—and they have also provided broader security and privacy perspectives pertaining to society and education. Additionally, I'm grateful to Holly Stearne, Joe Elsener, Susan Perry Cable, and the late Sandy Hall; they have each graciously helped me in some way over the course of my PhD, and their presence in Sudikoff have made Sudikoff more enjoyable. I am also grateful to Pat Berry who has been kind and supportive throughout my PhD. He's always been a friendly face in Sudikoff!

Finally, I would like to thank my parents, brother, relatives, and friends for their support and for generally putting up with me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis pursues the challenge of bridging the gap between security and privacy intentions and outcomes. Our primary focus is on helping security practitioners make decisions that serve their security goals. In this introductory chapter, we motivate our work, discuss our key contributions, and present a bird's-eye view of the thesis, wherein we discuss the other chapters of the thesis and explain how they fit together as a cohesive whole. Last, I explain my contributions to the published papers and research upon which this thesis is based.

Section 1.1

Motivations & Background

Security practitioners must often make decisions, such as choosing a password composition policy for a system or service, selecting security advice to administer to users, or choosing a mechanism to time users out of a system or a session. However, making a well-informed decision that produces the desired security outcome is often fraught with challenges. Well-intentioned security solutions may get in the way of workflow, driving well-intentioned users to develop and employ circumventions to get their jobs done—circumventions that may not only nullify envisioned security gains, but also produce unforeseen risks. Even if full user compliance is achieved, the security decision may introduce unanticipated workflow impediments that interfere with the user's primary task or other organizational objectives; in some cases, the security solution may be detrimental to other organizational goals, leading to its rollback. Cascading failures present another class of challenges: the lack of a feedback loop—or worse, a feedback loop that misinforms, widening the disconnect between what the security practitioner thinks is happening and reality—may negatively impact future decisions. Other challenges, such as those presented by regulatory constraints and legacy constraints, also must be considered. Given that these and other complexities muddy the waters, we aim to assist security practitioners in making sound decisions, specifically, ones that produce outcomes that align with their intentions.

Before diving into our work, let us briefly take a step back to set the context for our work by discussing related work: This thesis emerges at a time when there is newfound awareness in the security community that desired security outcomes are rarely realized by using textbook models that are incongruous with reality. These revelations are illuminated by much research conducted under the banner of HCISec, research guided by the belief that human-computer interaction lies at the heart of security, e.g., [178]. Many usability studies have pursued topics pertaining to user perceptions, user behaviors, security, usability, and circumvention; they have employed various methods such as holding focus groups, conducting usability experiments both in-person and on platforms like Amazon Mechanical Turk, and analyzing data gathered via logs and browser plug-ins, e.g., [4, 50, 193]. Models and simulations have been developed to explain, reproduce, and predict behavioral responses to security decisions, e.g., [63, 161, 171, 45]. Lessons and guiding principles for usable security have been developed, e.g., [16, 217]. And usable security positions have been advocated, e.g., [110, 185]. This, of course, only scratches the surface of existing usable security research.

The work presented in this thesis complements and builds upon the existing usable security literature. We focus on delivering knowledge, tools, and principles that can help people make well-informed decisions that realize their security objectives. That is, we pursue the overarching goal of bridging the gap between intent and outcome.

Section 1.2

Thesis Structure and Overview

This thesis contains six primary chapters woven together by the thread of empowering security practitioners to make more effective security-minded decisions that meet their objectives. Each chapter explores a single technique in pursuit of our grand objective of bridging the gap between intent and outcome. As such, we chose to discuss related work on a per-chapter basis, rather than having all the related work appear in one central location. Following the six primary chapters, we tie everything together and reflect on what we've learned in the concluding chapter of the thesis.

An overview of the thesis—its objectives and major chapter contributions—is provided pictorially in Figure 1.1.

Section 1.3

Chapter Synopses

We now provide synopses for the chapters that follow.

1.3.1. Chapter 2 Synopsis: Password Logbooks: Gleaning Usable Security Insights from Amazon Reviews

As research has shown, stringent password composition policies may backfire by frustrating users or driving them to circumvent recommended password practices, e.g.,

Thesis Objectives:

This thesis aims to improve security by bridging the gap between the intentions that guide one's decisions and the outcomes that those decisions ultimately lead to.

We seek to:

- provide insights into users' beliefs, goals, decisions, and behaviors
- conduct studies that reveal how users behave and the factors that influence those behaviors
- create models and simulations that shed light on the mismatch between intent and outcome

This thesis comprises six primary chapters, each of which demonstrates one technique that can help bridge the intent-outcome gap. We list the primary contributions of each chapter to the right. Following the six primary chapters, we reflect on our work and discuss common themes that emerge in the process of conducting our research.

Chapter Contributions:

Chapter 2: We analyze reviews of password logbooks, notebooks used to record passwords, available on Amazon. These reviews provide insights into user goals, user beliefs, user struggles in managing passwords, and a variety of coping strategies users employ.

Chapter 3: We create an agent-based simulation to study the impacts of adopting a collection of password policies across services. Such simulations may help in understanding user behaviors and responses to security policies, comparing policies, and detecting vulnerable points within systems.

Chapter 4: We explore the idea of supplementing the contributions provided by the field of language-theoretic security with an approach that captures the limits of what actual humans who are subject to various deficiencies---not idealized humans that are impervious to them---can do.

Chapter 5: We conduct a study that uses eye tracking to determine how users parse and classify URLs. The eye measurements provide reliable data on how users visually process information and also the underlying cognitive processes that drive those visual processes.

Chapter 6: We complement our eye-tracking study with an MTurk study where users are again tasked with parsing and classifying URLs. We explore how a variety of URL features and other factors affect URL classification.

Chapter 7: We build upon our earlier work where we used mismorphisms - a model based in semiotic triads - to capture circumvention scenarios. We present our recent work on developing a complementary logical model.

Figure 1.1: This thesis in a nutshell.

by choosing weak passwords, reusing existing passwords, writing down their passwords, or relying on the password reset mechanism to authenticate. This expectation-outcome mismatch has even led Bill Burr, an author of a 2004 NIST standard [41] that advocated for stringent password composition requirements on the basis of (a misapplication of) Shannon entropy, to denounce those very recommendations [116]. Indeed, the more recent standard has abandoned this approach altogether [71, 70]. Of course, stringent password composition polices are not the only drivers of user circumvention; the sheer number of accounts users must maintain, the frequency of mandatory password resets, account sharing needs, and myriad other factors encumber users and drive them to circumvent. While a user's decision to circumvent may not be tied to a single particular security decision, catalysts for circumvention, such as stringent password composition policies, often do stem from a security decision.[1] Collectively, the security-minded decisions made by security practitioners in pursuit of improving password security have failed to produce the desired results, in large part because the user does not conform to the assumed textbook ideal.

An abundance of literature examines the passwords that users construct, the efficacy of password policies, and how users manage passwords. However, less attention has been given to some of the more nuanced topics, such as the extent to which users attempt to engage in secure behaviors, their awareness of the security repercussions of their password management strategies, and their attempts to reduce or mitigate perceived repercussions. We sought to examine these understudied topics by examining password logbooks—notebooks specially designed for end users to record passwords

[1] Although password managers solve many of these problems, it's also important to note that password managers are not a panacea, e.g., password managers are not always an obvious solution for end users and the required time and mental energy to decide on whether to use a password manager—and, if so, which to use—often serves as a barrier to entry. There are also problems that many password managers simply do not address, such as sharing account credentials with family members. This is a limitation that has been experienced by a member of our broader research group and also one that was expressed in reviews of password logbooks.

and potentially other relevant information for computer use—that are available on Amazon. In addition to examining the password logbook market on Amazon, we employed grounded theory, a heavily used methodology from the social sciences for doing qualitative analysis, to generate and analyze a corpus of Amazon Verified Purchase reviews for password logbooks. The sheer existence and breadth of these notebooks speak to the struggles regular users have in managing their passwords, as well as the features they desire. Moreover, the product reviews for the password logbooks provide valuable insights into end user beliefs, concerns, and behaviors.

1.3.2. Chapter 3 Synopsis: Measuring the Security Impacts of Password Policies Using Cognitive Behavioral Agent-Based Modeling

Consider an employee tasked with creating password policies and other authentication policies for their organization. Or a member of a regulatory body charged with developing security regulations or compliance protocols for keeping personally identifiable information safe in hospitals. Or a security practitioner selecting a method to defend their service against denial-of-service attacks without blocking legitimate users. While it may be easy to make the “right” security decision in some circumstances, it can be quite difficult in others due to: misperceptions regarding users and their beliefs, goals, and limitations; system considerations; existing organizational policies; existing regulation; and other factors. This complexity often produces a mismatch between security projections and outcomes. However, security decisions *must* be made, creating a need for tools that assist in decision-making. We demonstrate how agent-based simulation can serve as such a tool by studying the use case of deciding upon a password composition policy.

An agent-based simulation involves an agent—a program that aims to simulate the behaviors of a human or other sentient being—that repeatedly takes actions according to some decision-making process within a given simulation environment. Our goal is to better understand proposed security solutions in the deployment context—their shortcomings, the concomitant workflow and usability issues, and the general effectiveness of the proposed solutions. We are interested in scenarios where user behavior is tightly linked to security outcomes; hence, it is vital to model humans as they truly are, constrained by memory limitations, emotion, misperceptions, and other factors that guide behavior. We therefore build our simulations atop *DASH* [31, 30], an agent-based modeling framework that's capable of capturing human factors like emotion, stress, cognitive burden, and workflow considerations within subsystems, which guide agent behavior.

The chapter discusses a password simulation we built with the goal of helping practitioners make password-related decisions—such as selecting a password composition policy (i.e., the rules for determining what constitutes a proposed password), selecting a password reset mechanism, and deciding whether to adopt mandatory password resets (and, if so, the frequency of these resets)—that best serve an organization's security objectives and other organizational objectives. Agents in this simulation simulate users who create accounts, use services, and attempt to comply with rules and recommendations. Moreover, we coded the agents to simulate human deficiencies, such as memory limitations and forgetting, as well as circumvention behaviors, namely writing down passwords, reusing passwords, and relying on password reset mechanisms instead of remembering them.

1.3.3. Chapter 4 Synopsis: Human-Computability Boundaries

The security of a protocol rests on its ability to operate only on expected input. The parser is the part of the protocol that is responsible for ensuring that the input conforms to the grammar that specifies acceptable input on which the protocol is *supposed to* run. Protocols are not intended to operate on input that does not belong to the language specified by the grammar. A key tenet of language-theoretic security

(LangSec) [2] that directly follows this line of reasoning is that the parser should run in full, only passing along input that has been recognized. That is, processing should only be performed on input that has already been recognized.

Another key tenet of LangSec is the principle of least expressiveness. It states that during protocol or parser construction one should use the least expressive grammar that will suffice. More precisely, one should ensure their chosen grammar lies within certain computability boundaries corresponding to the problems of Turing-decidability and parser equivalence. These boundaries are fitted to an extended version of the Chomsky hierarchy that differentiates between non-deterministic and deterministic pushdown automata. This extended hierarchy, like the 4-class Chomsky hierarchy that is usually presented to computer science students[2], is indeed a proper containment hierarchy. [166]. Of course, staying within these boundaries does not guarantee security. Rather, one should think of staying within the computability boundaries as a single article of evidence, albeit a critical one, in support of security. We pursue the identification of another key article of evidence.

As LangSec aims to understand and account for the limitations of machines, we seek to understand the limitations of humans as it pertains to securing and using protocols and parsers. Although initial conceptions of computation did involve *human computers* (e.g., see [47]), those conceptualizations abstracted away many of the limitations that many actual humans face in practice—finite and small memories, impatience, cognitive biases, bounded rationality, the dual-process model of cognition, and so forth. The sole focus on humans as computers also does not capture the many roles humans play regarding code. Actual humans design code, develop code, and use code—and there are problems lurking at every part of the code's lifecycle. We discuss approaches to developing a model to capture human-computability boundaries and

[2] Sipser [176] provides a wonderful discussion of the hierarchy from the computer-science angle.

how such a model can be merged with LangSec.

1.3.4. Chapter 5 Synopsis: Eyes on URLs: Relating Visual Behavior to Safety Decisions

There's a disconnect between the heuristics that users employ when parsing URLs and the information embedded within actual URL structure. Many phishing attacks exploit this mismatch. Such attacks are well documented in the literature, e.g., [55, 134]. However, to the best of our knowledge, there are no studies that use eye-tracking to learn how users truly parse URLs.

Eye tracking tells us about how users visually process information. Moreover, in the right circumstances, it also can reveal information about the underlying cognitive processes via pupillary response; in essence, pupils dilate when the cognitive load for a user is high and they contract when it is low. In this project, we examine how users determine the safety of a URL and what cues they use. We create a URL corpus comprising safe and unsafe URLs, the criteria of which we explain in the chapter. The experimental setup consists of users classifying a series of images of URLs on a computer screen as safe or unsafe by clicking on-screen buttons while wearing an eye tracker. Following the URL classification portion of the experiment, participants fill in a questionnaire. We disaggregate the URLs into components—primarily the scheme component, the authority component, and everything following the authority component—and study how users visually process each component.

1.3.5. Chapter 6 Synopsis: An MTurk Study Examining How Users Evaluate URLs

We report on preliminary findings from a study that is similar in spirit to the last one; however, we lose the eye tracker and instead conduct the URL classification study over Amazon Mechanical Turk. This allows us to examine more URLs, use a larger population of users, and explore a variety of different conditions, at the cost of losing the data on users' visual and cognitive processes, which would have been afforded to us by an eye tracker. We examine user susceptibility to URL redirection attacks [203], ASCII homograph attacks [206], combosquatting attacks, and more. We also examine how users perceive URL shorteners [212], gatekeeper URLs [118, 148], and domain names comprising words with negative, neutral, and positive valences, as well as the impact of font on URL classification.

1.3.6. Chapter 7 Synopsis: A Logic for Mismorphisms

When contemplating the ramifications of a security decision, understanding how similar security decisions have played out in the past can improve the accuracy of decision-maker projections and therefore help them to make a better-informed decision in the present. Of course, this is by no means a silver bullet, as we've argued in the past [33]. Not all organizations are the same. Context matters and inappropriate reliance on outcomes of past events can indeed lead to worse decisions. That said, used wisely, past information can also be quite valuable so long as the relevant context is clearly and effectively communicated. To this end, we share our work on developing a model to capture the underlying causes of security issues. We contend that cataloging security issues of the past by their underlying causes can help inform security decisions of the present.

In earlier work [180] , we sought to catalog and explain the underpinnings of security problems seen in practice, which so often stem from differential representations of reality, e.g., user and security practitioner representations, the system representation, the actual reality. Roughly, we call these differential representation *mismorphisms*. An inquiry into earlier work in semiotics and, more specifically, the semiotic triads presented by Ogden and Richards [133], led us to a natural model for expressing these mismorphisms and the events they induce. Using said model, we catalogued and classified numerous security problems.

We expand upon our prior work by capturing mismorphisms using a logical model. We also demonstrate how this model allows us to capture the causes of a variety of security problems. The end goal of this work is to systematize the knowledge (such as the case study performed by Heckle [76] or scenarios from the Risks Digest periodical [127]) available in the academic literature and elsewhere in an effort to inform security and privacy decisions.

1.3.7. Chapter 8 Synopsis: Conclusion

In the concluding chapter, we review the previous chapters and provide key takeaways. We then provide a short discussion of themes that emerged during our research. Finally, we provide directions for future work and conclude.

Section 1.4
Acknowledgments

This thesis revises and extends text from prior publications. Thus, for this section, we temporarily drop the welcoming *we* in favor of the awkward *I* as I list these publications, explain where they fit within the thesis, and state my contributions to them. Each of these papers involved a collaboration among researchers. My focus in this section is primarily on explaining *my* contributions. I hope that it is clear that, in the interest of brevity, I have not exhaustively stated the many contributions my coauthors have made to these papers.

Chapter 2 Acknowledgements

Chapter 2 is based on:

> Vijay Kothari, Jim Blythe, Ross Koppel, and Sean Smith. Password Logbooks and What Their Amazon Reviews Reveal About Their Users' Mo-

> tivations, Beliefs, and Behaviors. In *2nd European Workshop on Usable Security (EuroUSEC 2017)*. IEEE, 2017

My contributions to this paper involved collecting data, doing data analysis, generating figures, and writing a significant portion of the paper. Ross Koppel helped with applying grounded theory to qualitatively analyze the data. My coauthors provided feedback throughout the study, and they helped in writing and refining the paper.

We note the paper's acknowledgments:

"This material is based upon work supported by the Maryland Procurement Office under Contract No. H98230-14-C-0141. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Maryland Procurement Office. Koppel's work was supported in part by NSF CNS-1505799 & the Intel-NSF Partnership for Cyber-Physical Systems Security & Privacy."

Chapter 3 Acknowledgements

Chapter 3 is based on:

> Vijay Kothari, Jim Blythe, Sean W Smith, and Ross Koppel. Measuring the Security Impacts of Password Policies Using Cognitive Behavioral Agent-Based Modeling. In *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, page 13. ACM, 2015

My contributions to this paper involved creating the simulation, conducting tests using the simulation, creating plots, and writing a significant portion of the paper. Jim Blythe helped me get up to speed with the agent-based simulation framework *DASH* and provided suggestions on creating the simulation. My coauthors provided feedback throughout the whole development of the simulation, and they helped in writing and refining the paper.

We note the paper's acknowledgment:

"This material is based upon work supported by the Maryland Procurement Office under Contract No. H98230-14-C-0141."

I should also briefly note some follow-up research based on this paper that I have contributed to but which were led by undergraduate researchers; these papers focused on further validating the simulation [99] and using it to compare the efficacy of password construction and memorization techniques [128]. The reader who is interested in this simulation may want to refer to those papers as they are more recent. However, the two papers are beyond the scope of this thesis. We do not discuss them beyond this point.

Chapter 4 Acknowledgements

Chapter 4 is based on:

> Vijay Kothari, Prashant Anantharaman, Ira Ray Jenkins, Michael C. Millian, J. Peter Brady, Sameed Ali, Sergey Bratus, Jim Blythe, Ross Koppel, and Sean W. Smith. Human-Computability Boundaries. In *Security Protocols Workshop XXVII (To Appear)*. Springer International Publishing, 2020

I conceived of many ideas expressed in this paper, created the figure used in the paper, and made significant contributions to the writing of the paper. My coauthors helped write and refine the paper. And they provided valuable feedback.

We note the paper's acknowledgment:

"This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-16-C-0179 and Department of Energy under Award Number DE-OE0000780.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United

States Air Force, DARPA, United States Government or any agency thereof."

Chapter 5 Acknowledgements

Chapter 5 is based on a paper soon to appear in ETRA 2020:

> Niveta Ramkumar, Vijay Kothari, Caitlin Mills, Ross Koppel, Jim Blythe, Sean Smith, and Andrew L. Kun. Eyes on URLs: Relating Visual Behavior to Safety Decisions. In *Proceedings of the 12th ACM Symposium on Eye Tracking Research & Applications (To Appear)*, Stuttgart, Germany, 2020. Association for Computing Machinery. doi: 10.1145/3379155.3391328

My contributions to this paper primarily involved generating the URL corpus, creating some figures and tables, doing a small amount of data analysis, working on the questionnaire, and writing a portions of the paper. However, it's worth stressing that this was a highly collaborative and iterative project, and we ran each aspect of the project by one another. Although we collaboratively decided on what the experimental setup and the user interface for the URL classification task should be, I had no role in the actual setup of the experiment, nor did I develop the user interface; these were done by Niveta Ramkumar and Andrew Kun. Niveta Ramkumar and I roughly split the work of writing most of the paper in close consultation with Andrew Kun, with other colleagues providing critical additions and revisions. Much of the writing was done jointly at the same time on Overleaf. That said, where greater technical knowledge was required, notably in the writing of related work, Niveta Ramkumar focused more on the eye-tracking and mood induction segments, whereas I focused a bit more on the security segments. Ross Koppel helped significantly with devising the post-task questionnaire. Caitlin Mills helped us with mood induction. All coauthors provided valuable feedback and helped in writing and refining the paper.

We note the paper's acknowledgment:

"The work of Ramkumar and Kun was supported in part by NSF grant OISE 1658594."

Chapter 6 Acknowledgements

While Chapter 6 is not based on any published work, we expect we will soon publish a paper based on this work. I played a major role in coming up with the ideas and conditions explored in the study, creating the URL corpus, doing data analysis, adapting the questionnaire from the eye-tracking study, and writing the content presented in the chapter. I also helped with designing and testing the Qualtrics interface. However, I did not create the URL images, and I only minimally contributed to setting up the task on MTurk by giving feedback to my coauthor, Prashant Anantharaman. Ross Koppel, Sean Smith, and Jim Blythe provided lots of valuable feedback on the method, the URL classification task, and the post-task questionnaire.

Chapter 7 Acknowledgements

In Chapter 7, we review our earlier work, though we do not borrow text from it:

> Sean W Smith, Ross Koppel, Jim Blythe, and Vijay Kothari. Mismorphism: a Semiotic Model of Computer Security Circumvention. Technical report, Dartmouth College, Department of Computer Science, 03 2015

I was not the driver of this work, nor do I feel my contributions were sufficiently large that it warrants presenting this paper as a primary thesis contribution. And I don't. We do not borrow text from the paper without quotation marks and a citation—and we treat this paper as prior work that is instrumental in understanding our more recent work.

We note the paper's acknowledgment:

"This material is based in part upon work supported by the Army Research Office under Award No. W911NF-13-1-0086."

Chapter 7 is based on:

> Prashant Anantharaman, Vijay Kothari, J. Peter Brady, Ira Ray Jenkins, Sameed Ali, Michael C. Millian, Ross Koppel, Jim Blythe, Sergey Bratus, and Sean W. Smith. Mismorphism: The Heart of the Weird Machine. In *Security Protocols Workshop XXVII (To Appear)*. Springer International Publishing, 2020

My contributions to this paper involved conceiving the idea for the paper, developing the logical formalism, writing significant portions of the paper (mostly in Sections 1, 2, 3, and 6), and helping to brainstorm mismorphism classes and entries for the catalog. I contributed only minimally to the actual writing of Sections 4 and 5. Prashant Anantharman wrote much of Sections 4 and 5. My coauthors provided valuable feedback; they also helped in writing and refining the paper.

We note the paper's acknowledgment:

"This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-16-C-0179 and Department of Energy under Award Number DE-OE0000780.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force, DARPA, United States Government or any agency thereof."

We also heavily use work from:

> Vijay Kothari, Jim Blythe, Sean Smith, and Ross Koppel. Data Privacy and the Elusive Goal of Empowering the User. In *Workshop on Moving Beyond a 'One-Size-Fits-All' Approach: Exploring Individual Differences in Privacy, CHI*, 2018

This is a position paper, which I wrote much of. My coauthors provided valuable feedback; they also helped in writing and refining the paper.

Chapter 8 Acknowledgements

The final chapter, Chapter 8, does not directly borrow text from any published work. The key takeaways mentioned in the chapter stem from my work and interactions with my PhD advisor, Sean Smith, as well as Jim Blythe, Ross Koppel, and Sergey Bratus. Some ideas expressed in the chapter overlap with the following position paper, which I helped to write; however, no text was directly used from the paper:

> Jim Blythe, Vijay Kothari, Sean Smith, and Ross Koppel. Usable Security vs. Workflow Realities: Work in Progress. In *Workshop on Usable Security (USEC 2018)*, 2018

We note the paper's acknowledgments:

"This material is based upon work supported by the Maryland Procurement Office under Contract No. H98230-14-C0141. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Maryland Procurement Office. Koppel's work was supported in part by NSF CNS-1505799 and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy. Blythe's work was supported in part by the Science and Technology Directorate of the United States Department of Homeland Security under contract number HSHQDC-16-C-00024. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security or the US Government."

Other Acknowledgments

This thesis is partly informed by the following book chapter that my colleagues and I had written:

> Prashant Anantharaman, J. Peter Brady, Ira Ray Jenkins, Vijay H. Kothari, Michael C. Millian, Kartik Palani, Kirti V. Rathore, Jason Reeves, Rebecca Shapiro, Syed Tanveer, Sergey Bratus, and Sean W. Smith. Intent as a Secure Design Primitive. In C.A. Kamhoua, L.L. Njilla, A. Kott, and S.S. Shetty, editors, *Modeling and Design of Secure Internet of Things (To Appear)*, chapter 24. Wiley, 2020. ISBN 9781119593362. URL `https://books.google.com/books?id=fljywAEACAAJ`

While we do not borrow text from the book chapter, the general focus on intent for conceptualizing how security problems arise is very relevant to this thesis. My contributions to the book chapter were primarily in writing portions of the sections entitled “Introduction” and “A LangSec Primer”.

We note the paper’s acknowledgments:

“This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-16-C-0179, ONR under grant N00014-18-1-2022 and Department of Energy under Award Number DE-OE0000780.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force, DARPA, ONR, United States Government, or any agency thereof.”

Chapter 2

Password Logbooks: Gleaning Usable Security Insights from Amazon Reviews

The existence of and market for *password logbooks*, notebooks designed primarily for recording password information, illuminates a sharp contrast: what is often prescribed as proper password behavior—e.g., never write down passwords—differs from what many users actually do. These password logbooks and their reviews provide valuable insights into their users' beliefs, motivations, and behaviors. We examine these password logbooks and analyze, using grounded theory, their reviews, to better understand how users think and behave with respect to password authentication. Several themes emerge including: previous password management strategies, gifting, organizational strategies, password sharing, and dubious security advice. Some users argue these books enhance security.

Section 2.1

Introduction

User behavior often conflicts with advice and policies prescribed by security practitioners. To name a few examples of such behavior:

- users write down passwords on sticky notes and affix them to computers,
- users use the same password for different services, and
- users ignore certificate warnings.

Recognizing and understanding such behavior is critical to improving security solutions. More generally, better understanding of user motivations, perceptions, constraints, and behaviors empowers security practitioners both to select more effective security policies and mechanisms and to offer better security guidance, which increases user compliance and mitigates the risks posed by circumvention, ultimately improving both individual and aggregate security.

Security decisions based on false assumptions—assumptions stemming from disconnects between certain aspects of users and what security practitioners believe about users—will almost always be ineffective. Thus, it is imperative to learn what users do and why they do it, and then to tailor security policies, security mechanisms, and security advice based on this understanding. Indeed, this has been a major aim of usable security research, much of which relies on more traditional, controlled data acquisition methods, such as surveys and behavioral experiments.

In this chapter, we build on and complement existing research by studying the numerous password logbooks, notebooks designed for users to record passwords and other information, that are available on Amazon.[1] We also analyze their reviews.

[1] These are also known by other names, e.g., password notebooks, password journals

Of the several hundred password logbooks available on Amazon, we examine 116 unique password logbooks, and we analyze 4,330 unique reviews for them. These reviews provide remarkable insights into reviewers' motivations, pre-purchase and post-purchase behaviors, and perceptions and misperceptions about security, among other findings.

We discuss related work in Section 2.2 and then provide an overview of our study in Section 2.3. We analyze password logbooks and their reviews in Sections 2.4 and 2.5. In Section 2.6 we discuss our findings. We detail our methodology and note both limitations and advantages of the approach in Sections 2.7 and 2.8. We conclude with suggestions for future work in Sections 2.9 and 2.10.

Section 2.2 Related Work

Gaw and Felten [65], as well as other researchers, studied password management strategies, such as writing down passwords on paper or sticky notes and reusing the same password or small variations of a core password across services. Scholars have commented negatively on the use of dedicated logbooks to record passwords and they have also expressed the view that writing down passwords is a poor security practice that reflects the unusability of authentication processes, e.g.,[75], [4, 82]. While conventional wisdom and many security experts deplore the practice of writing down passwords, many experts have also advocated such practices so long as the passwords are securely stored, e.g., [136]. Irrespective of whether these practices are secure, researchers have shown the viability and rationality of user adoption of such practices, e.g., Herley [78] showed that many "incorrect" password management strategies users employ are rational.

Many researchers have used grounded theory and other methods to better understand user password decisions and behaviors. Grounded theory is an iterative qualitative research methodology for discovering theories that emerge directly from the data. [205] Stobert and Briddle [183] interviewed users to learn how they manage their account credentials. They then applied grounded theory to explain the password lifecycle, i.e., the behaviors users employ to keep track of a password throughout its use. Fagan and Khan [59] conducted a survey on Amazon Mechanical Turk to understand why users make security-related decisions. Inglesant and Sasse [85] gave users a diary to record their password behaviors for a week and conducted interviews afterward, findings that users want to comply with security policies but struggle to do so. They suggested policies should be designed using HCI principles.

Ha and Wagner [73], have used product reviews to learn more about user behaviors, perceptions, and attitudes. Alkadi and Renaud [7] analyzed user reviews of password managers on the Google Play Store and the iTunes App Store, and they conducted a survey to understand user attitudes. We also analyze user reviews, but we do so on a larger scale with a significantly different subpopulation of users who circumvent often recommended security practices by using password logbooks.

Our work builds on previous efforts by a focused analysis of reviews and development of a typology of explanations (hereafter called *themes*). We illustrate each theme with examples from the products, their marketing, and their reviews. Researcher have employed automatic methods to analyze user reviews for products other than password logbooks, e.g., [96, 80]. We, however, use methods from grounded theory that blend manual and computerized text analysis to extract themes from reviews.

Section 2.3

Study Overview

We define a *password logbook* as any printed book marketed for users to record passwords and related account information (e.g., names of services, usernames, security hints), as well as other computer and internet-related information (e.g., network settings, ISP telephone numbers).

We create a data set comprising password logbooks that have one or more Amazon Verified Purchase reviews, along with their reviews.[2] The final dataset comprises 116 password logbooks and 4,330 reviews for them with duplicate reviews removed. We analyze the products and used grounded theory methods to inductively construct common themes in the reviews using two coders. A complete discussion of the methodology and limitations is provided in Sections 2.7 and 2.8.

Section 2.4

Product Findings

We reviewed 116 password logbooks. The most-reviewed book had 1,811 reviews, of which 1,687 were Verified Purchase reviews; on the other end of the spectrum, many password logbooks in our set had only a single Verified Purchase review. Indeed, as seen in Figure 2.1, a few password logbooks accounted for a large fraction of reviews. Once duplicate reviews were removed, we found that the first five products accounted for 2,973 of the 4,330 reviews or equivalently, 68.7% of the reviews.[3]

Figure 2.2 is a histogram of reviews by review date. As we gathered the final

[2] "An 'Amazon Verified Purchase' review means [Amazon] verified that the person writing the review purchased the product at Amazon and didn't receive the product at a deep discount." [1]

[3] These numbers depend on which of the duplicate reviews to remove or rather, more precisely, which review of a collection of identical reviews for different books to keep. Still, as there were only a few duplicate reviews, the numbers would only vary slightly (less than 1%) depending on this choice. Please see 2.7 for further details on how we handled duplicate reviews.

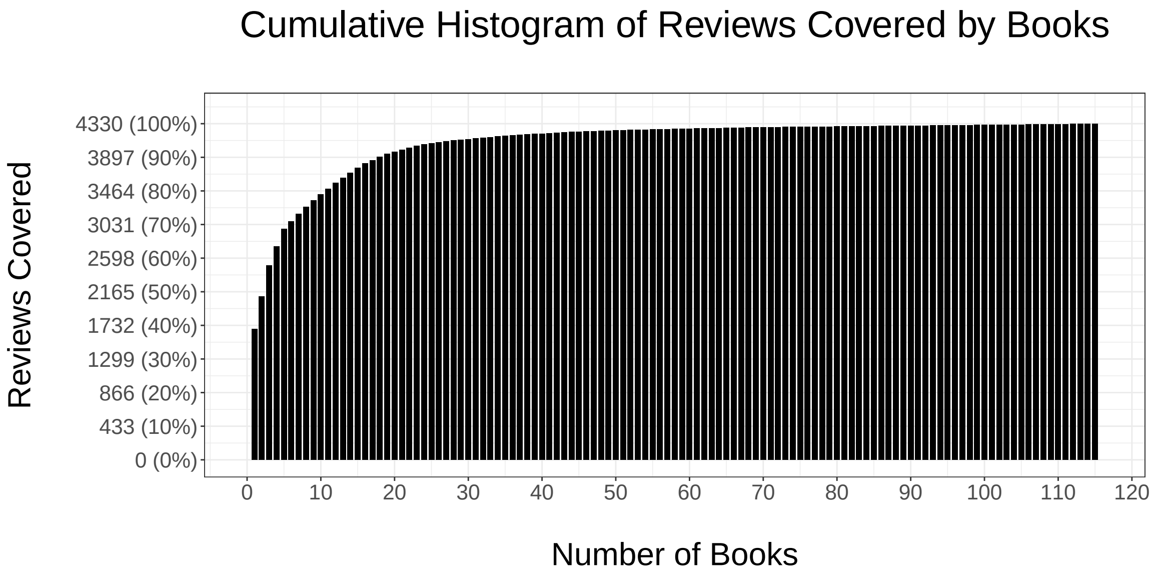


Figure 2.1: Cumulative histogram of reviews covered by books. This histogram shows the number of reviews covered by a subset of books, selected in non-increasing order of number of reviews. For example, the graph shows that the 10 most reviewed password logbooks account for 3,418 (78.9%) of the reviews.

set of reviews on March 7, 2017, we collected only a fraction of the reviews from 2017. Therefore, we derived a projection for the total number of reviews in 2017 by scaling the number of reviews we had seen in 2017 by the number of reviews posted in the previous three years over the fraction of reviews posted before or on March 7 in the previous three years. The graph reveals that password logbooks listed on Amazon have received more reviews in recent years. This may be due to a number of factors, e.g., more demand for password logbooks, more password logbooks available on Amazon, and people opting to buy books via online stores like Amazon instead of brick and mortar stores.

Password logbooks were generally highly rated with the average rating being 4.56 out of 5. As a few popular books covered most reviews, this is expected.

Front covers of ten of the password logbooks appear in Figure 2.3. Additionally, pictures from the interiors of four password logbooks are provided in Figure 2.4.

Our analysis revealed a number of features that differentiate the password log-

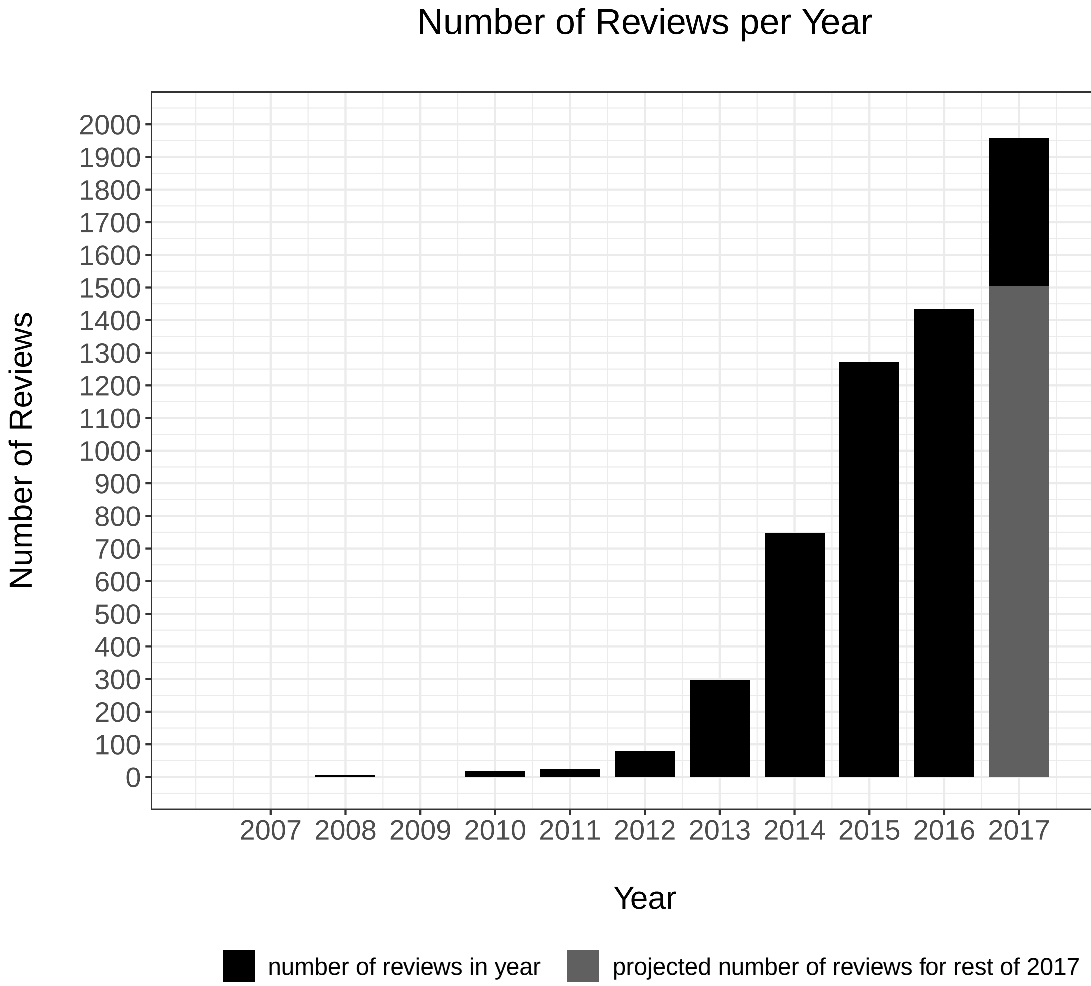
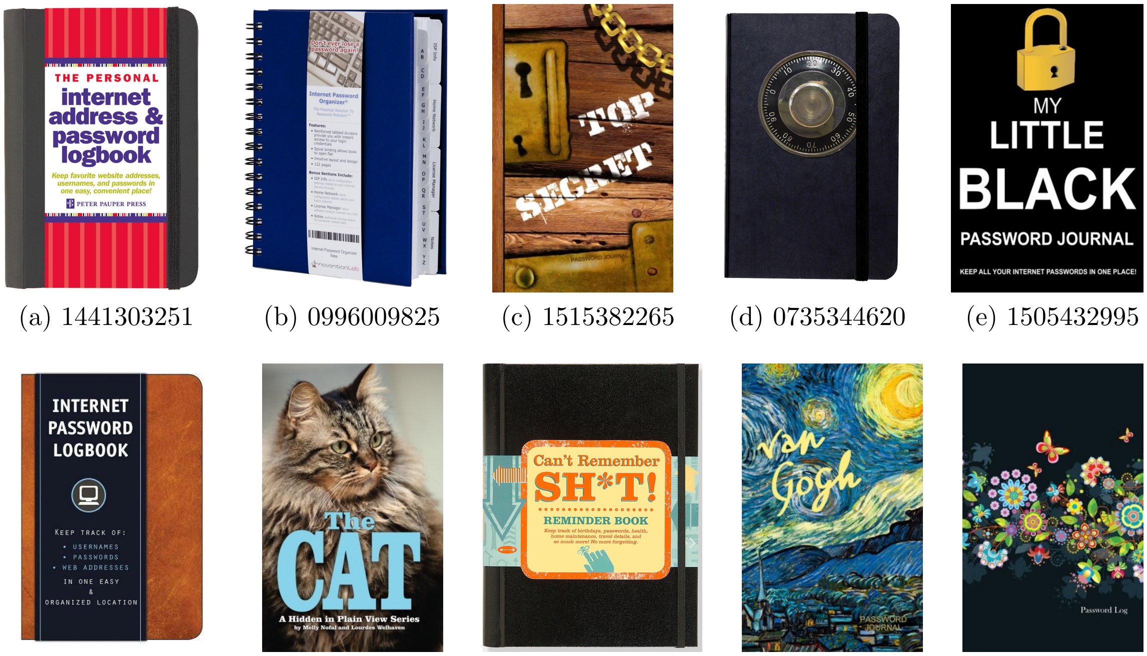


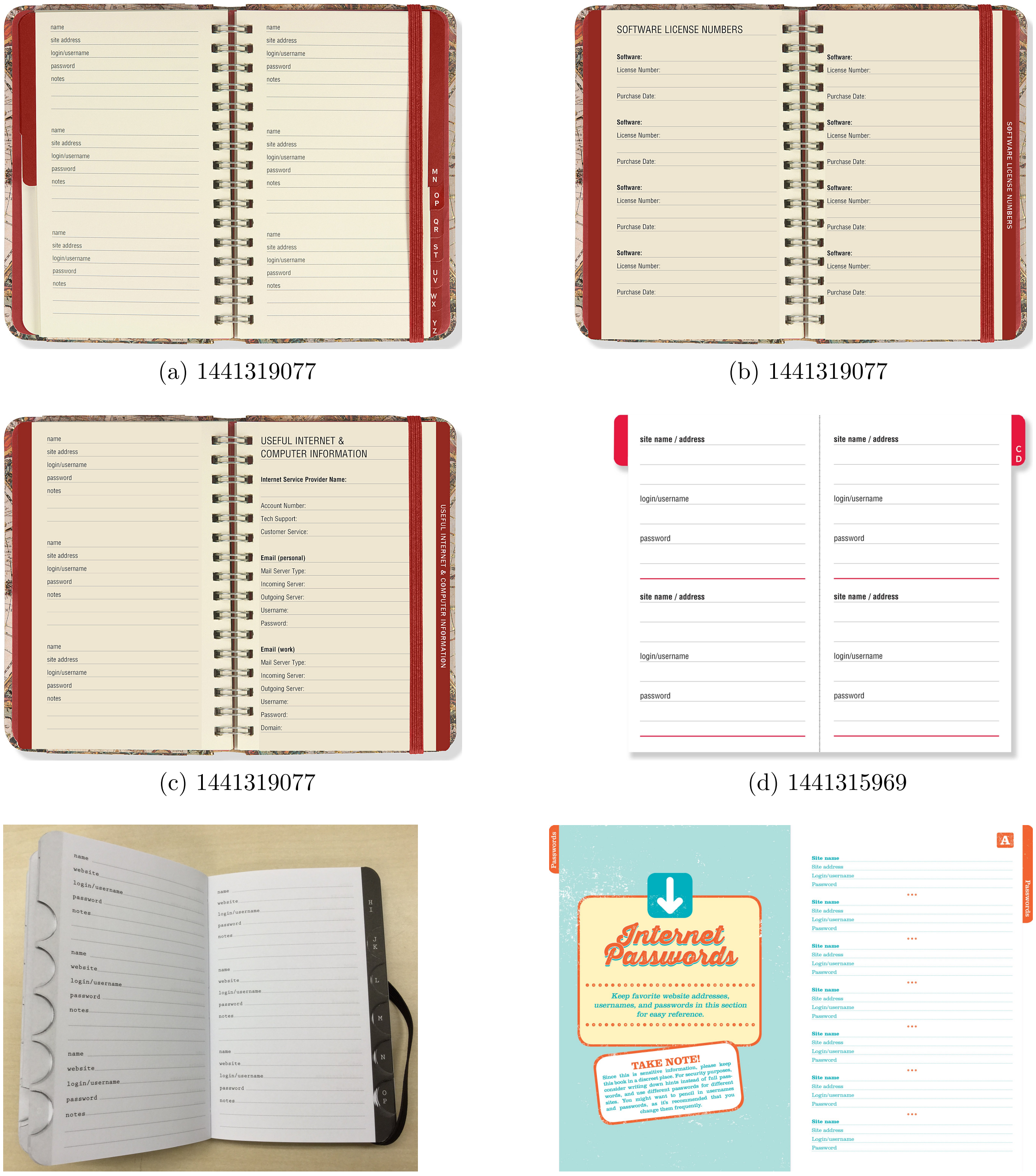
Figure 2.2: Number of Amazon Verified Purchase reviews per year. We collected reviews on March 7, 2017. The projected number of reviews for 2017 was obtained by multiplying the number of reviews seen in 2017 by a scaling factor; this scaling factor is the number of reviews in the years 2014–2016 divided by the number of reviews before or on March 7 in 2014–2016. The project number of reviews for the rest of 2017 was obtained by subtracting the number of reviews already observed from the projected total.


THE PERSONAL
internet address & password logbook
Keep favorite website addresses, usernames, and passwords in one easy, convenient place!
PETER PAUPER PRESS
Don't ever lose a password again!
Internet Password Organizer
TOP SECRET
MY LITTLE BLACK PASSWORD JOURNAL
KEEP ALL YOUR INTERNET PASSWORDS IN ONE PLACE!
INTERNET PASSWORD LOGBOOK
KEEP TRACK OF:
USERNAMES
PASSWORDS
WEB ADDRESSES
IN ONE EASY & ORGANIZED LOCATION
The CAT
A Hidden in Plain View Series
Can't Remember SH*T!
REMINDER BOOK
Van Gogh
PASSWORD JOURNAL
Password Log

(a) 1441303251 (b) 0996009825 (c) 1515382265 (d) 0735344620 (e) 1505432995

(f) 1631061941 (g) 1500863548 (h) 1441319441 (i) 152372398X (j) 1515246825

Figure 2.3: Front covers of 10 of the 116 password logbooks examined in this work. Each image is located at https://amazon.com/dp/ASIN/ (ASINs specified in subcaptions).


name
site address
login/username
password
notes
M N
O P
Q R
S T
U V
W X
Y Z
SOFTWARE LICENSE NUMBERS
Software:
License Number:
Purchase Date:
USEFUL INTERNET & COMPUTER INFORMATION
Internet Service Provider Name:
Account Number:
Tech Support:
Customer Service:
Email (personal)
Mail Server Type:
Incoming Server:
Outgoing Server:
Username:
Password:
Email (work)
Domain:
site name / address
C D
website
H I
J K
L
M
N
O P
Passwords
Internet Passwords
Keep favorite website addresses, usernames, and passwords in this section for easy reference.
TAKE NOTE!
Since this is sensitive information, please keep
this book in a discreet place. For security purposes,
consider writing down hints instead of full pass-
words, and use different passwords for different
sites. You might want to pencil in usernames
and passwords, as it's recommended that you
change them frequently.
A
Site name
Site address
Login/username
Password

(a) 1441319077

(b) 1441319077

(c) 1441319077

(d) 1441315969

(e) 0735344620

(f) 1441319441

Figure 2.4: Images of interiors of 6 of the 116 password logbooks we examined in this work. The first 3 images display pages from different sections of the same book. Each image is located at https://amazon.com/dp/ASIN/ (ASINs specified in subcaptions).

books:

- *Inconspicuousness:* Would an adversary not be able to recognize a password logbook as such? Some password logbooks had non-removable covers that said "password logbook" or had other indicators that enable people to easily identify it as a password logbook when closed. Others had similar covers and labels, but they could be easily removed and were intended to be removed. Some other books went one step further in that they masqueraded as a novel (e.g., see Figure 2.3g).

- *Password Security Tips:* Password logbooks provided various password security and book usage tips regarding keeping the password logbook in a safe place and not traveling with it, writing down password hints instead of passwords, not sharing passwords with others, using a pencil so passwords can easily and neatly be erased, and so forth. Some even gave instructions on how to create a strong password. Some tips contradicted the design and marketing of other password logbooks, even ones sold by the same vendor. For example, one password logbook advised the user not to travel with the book; however, the same vendor was selling a password logbook that was marketed as pocket-sized.

- *Durability:* Books varied in the durability of the binding, flimsiness of the cover, page thickness and ability to withstand ink and erasures, and other factors.

- *Aesthetics:* Books had a variety of different designs. A few books had unique aesthetics to target select demographics (e.g., children, women) and state such in their descriptions. For an example, the book seen in Figure 2.3c was marketed as "a fun kids' password journal with 'Top Secret' and 'Keep Out' on the front and back covers."

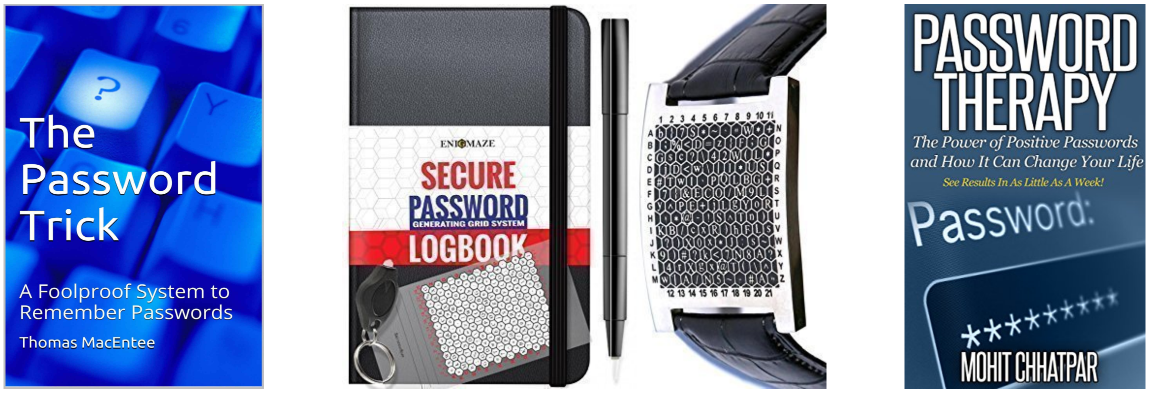
- *Size:* Books ranged in dimensions from 2.875” x 4.75” to 6.5” x 8.5”. In general, smaller books could easily fit in pockets, purses, and briefcases, whereas larger books provided more space and were easier to read.

- *Tabs:* Some password logbooks had tabs that allowed the user to more quickly find their passwords by service name. Many books devoted the same number of pages to every pair of consecutive letters, corresponding to a tab, though other tab layouts existed. Most users appreciated tabs, but some were frustrated due to a misalignment between the number of pages dedicated to tabs and user needs, granularity of tabs, durability of tabs, and visibility of tabs (some tabs protruded for greater visibility).

- *Elastic Band:* Some logbooks had an elastic band attached to the back cover to keep track of the owner’s place during use and to keep the book shut during non-use. An example of such an elastic band can be found in Figure 2.3d.

- *Contact Information:* Some books had space for the owner to enter their name, email address, and phone number. Of course, in the event that the password logbook is misplaced or lost, if this information is filled in, it may pose an additional privacy risk.

- *Other Entries:* Password logbooks ranged significantly in what information they allowed users to record. All password logbooks allowed users to record basic account credentials, i.e., site name, username, and password. However, many also allowed for other password-related information, e.g., multiple password entries per service with attached date fields, password question answers, notes. Moreover, many books had space to record other information that might be important to a computer user , e.g., home network information, software license keys. Figure 2.4 provides a few examples of entries within these books.


The Password Trick
A Foolproof System to Remember Passwords
Thomas MacEntee
ENIGMAZE
SECURE PASSWORD
GENERATING GRID SYSTEM
LOGBOOK
PASSWORD THERAPY
The Power of Positive Passwords and How It Can Change Your Life
See Results In As Little As A Week!
Password:
****
MOHIT CHHATPAR

(a) 2136504160 (b) B01I94N9TC (c) B00REGSI6G

Figure 2.5: A few other password-related products.
Each image is located at https://amazon.com/dp/ASIN (ASINs specified in subcaptions).

Numerous other password-related products are also available on Amazon, but they fall outside the scope of our study. Nevertheless, we briefly mention them here for completeness. These products include electronic password storage devices, books that give tips on creating and remembering passwords (including a unique flavor of self-help book entitled “Password Therapy”; see Figure 2.5a), books that suggest how to organize one’s records, including passwords, and alternative password management solutions. A few of these products are provided in Figure 2.5.

Section 2.5

Themes

Our analysis revealed numerous themes:

2.5.1. Love This Book!

Reviewers were often joyous about the password logbooks they purchased. Some reviewers wished that they had known about password logbooks sooner. Others used words and phrases such as “essential,” “vital,” and “can’t live without this” to express, often hyperbolically, their love for their password logbook.

Many reviewers considered the use of password logbooks (and similar circumventions) as an inescapable risk or reasonable tradeoff. Some argued that the requirement to track associations among services, usernames, and passwords, along with answers to security questions and other challenges (e.g., complex password composition requirements and frequent password resets) was overwhelming and that a password logbook was the best solution available.

2.5.2. Inconspicuousness

Reviewers generally valued inconspicuous password logbooks and, conversely, disparaged conspicuous ones. Some password logbooks had jackets or labels that said "password" on them. Some of these conspicuous covers were easily removable, but some others were not, which frustrated users. For example, one reviewer wrote:

> "It would be great if it didn't say 'Password Log' on the cover."

Some password logbooks resembled novels and blended in with other books. Reviewers generally found this clever. In reviewing a password logbook that masqueraded as a novel about a cat, a reviewer wrote:

> "No one thinks to look on the bookshelf or in a cat book for passwords."

2.5.3. Gifting & Spread of Circumventive Behavior

Numerous reviewers purchased or planned to purchase additional password logbooks for friends and family. Some purchases were gifts based on projected utility, whereas others were made upon request. Also, some reviewers mentioned that they purchased password logbooks after they saw friends or family use them. One reviewer wrote:

> "My mom bought one first and I saw how useful it can be so I got one too."

Gifting of password logbooks can be viewed as a way of spreading circumventive password behavior. This extends previous work that finds users obtain security advice from friends, families, and coworkers (e.g., [158, 151]). It also corroborates our earlier findings in enterprise settings [32]. That is, it's insufficient to only consider security advice prescribed by the enterprise; rather, it's just as important to consider security advice and behaviors spread by co-workers, family, friends and other enterprises.

2.5.4. Maintaining Passwords for Family Members

Reviewers explained that they used their password logbooks to keep track of their family members' passwords. For example, one reviewer wrote:

> "How about when your elderly parents keep having to change their password because they swear they are putting in the right one but it's not working... Yes, I put my [mom's] and [dad's] passwords in too, plus I did buy my mom one."

Another reviewer mentions:

> "Bought this for my father. Love how it is alphabetical. He was recently in the hospital and I found 3 sheets of ripped paper/notes with all his internet sites and passwords...some listed 2 or 3 times with different passwords. I had to take over bill paying while he was sick and this is working like a charm."

Another wrote:

> "I'm trying to get everyone organized. This is for my mother so I can find her passwords when she gets into trouble on the computer and I have to try and fix it. Before she had a confused and garbled note pad."

Indeed, many reviewers were concerned about how their family would get by in the event that they were no longer accessible. For example, one reviewer wrote:

> "If I were unavailable for any reason, my husband can now get into all the accounts for our kids activities, and not miss a beat! He can also get into all our bill paying areas if there is ever an issue. Must be kept under lock and key, but has given me a piece of mind!"

Another reviewer wrote:

> "Even though I use a password manager, I used this book in case something happens to me, my kids can get to the accounts."

This last quote is particularly interesting because the reviewer uses a password manager, which is often stated to be a good password management strategy, but keeps a password logbook as well.

Reviews also suggested that these concerns were prompted by life experiences. One reviewer stated:

> "I'm sure this will also be useful for the dreaded 'just in case' moment. A friend of mine's husband passed away a few years ago. To this day I don't know if she was ever able to access any of his sites on his computer because she didn't know any of his passwords. Always something to think about."

Another wrote:

> "Great little logbook to have handy. My husband recently passed away and I had a hard time finding a couple of things. This made me realize just how much I handled of the household finances and things. If something

> happened to me, my son would be left trying to [decipher] my mess. Keeps things organized and in one place and easy to secure where no one can stumble across it if need be."

2.5.5. Repeat Purchases and Multiple Logbooks

Some reviewers stated they had purchased a password logbook prior to the one for which they were writing a review. Reasons for doing so included: the previous password logbook lacked durability, the previous one lacked sufficient entries to store all passwords, and the reviewer wished to keep two or more password logbooks in different locations, e.g., one at home and one at work.

2.5.6. Age

Indicators of age were prevalent in a number of reviews. Reviewers often used old age and perceived memory loss as justification for using password logbooks, e.g., one reviewer wrote:

> "We are seniors with short term memory loss."

In contrast, as we noted earlier, some password logbooks were designed for children and were marketed as inculcating good security habits.

2.5.7. Size, Portability, and Storage

Many reviewers commented on the size and portability of password logbooks. Some reviewers preferred smaller, easily transportable books. Others preferred larger books capable of holding more passwords. Similarly, there was a tradeoff with the font size between readability and quantity of passwords that could be stored within the book.

Some reviewers routinely carried their password logbook in a briefcase, purse, or other carrying bag. Some left them on top of their desk or in their desk drawer.

Others made an effort to keep them in a safer place, e.g., a lockbox. These behaviors pertaining to carrying and storing the password logbooks often affected user preferences of the size of the password logbook.

2.5.8. Organization and the Centrality of Digital Life

Many reviewers stated these password logbooks helped them organize their accounts. In addition to just website addresses and passwords, users sought books that allowed them to store other information to access their accounts, e.g., usernames, answers to security questions. For example, one reviewer wrote:

> "There is not enough room for related information to passwords such as secret codes and question."

Another said:

> "[This is] perfect for those of us who are either brave to risk our information by signing up for numerous websites and we can't remember the password nor the website because there were so many, and for those that are new to the internet age and can't remember their name let alone a password to the only website they signed up for, this is the perfect book to use."

Moreover, many reviewers stated they used password logbooks as major organizing tools for their lives and their families. The books became a centerpiece of critical information about all of their accounts, wills, addresses, and other essential information.

2.5.9. Alternative and Previous Password Management Strategies Inadequate

A number of reviews reveal that alternative password management strategies, whether classified under the umbrella of circumvention or not, were inadequate. For example, one reviewer wrote:

> "Usually I would have just kept them in a file on a flash drive but....well...we did that and it got [corrupted] and now there are 4 accounts I am still trying to have shut down cause I don't remember ANY of the info I used to start the account."

Reviewers eagerly shared their previous password management strategies. These included writing down passwords on sticky notes, index cards, backs of envelopes, scattered sheets, and scraps of paper. More organized solutions included: storing passwords in an envelope containing paper scraps, a binder containing sheets of paper, and notebooks; storing passwords in text files and Excel spreadsheets; storing passwords on phones; and, as noted earlier, storing passwords on flash drives.

2.5.10. Risks

Many reviewers acknowledged the risk of keeping a password logbook, specifically, that it could be lost or stolen and wind up in the hands of an unscrupulous character. However, most, but not all, believed that password logbooks were better than other password management strategies. A few subthemes emerged here:

Perception that Password Logbooks Improve Security. Some reviewers suggested that even though password logbooks pose a risk, alternative password management aids pose an even greater risk, while not using any aid would cause them to struggle with remembering passwords, driving them to reuse passwords or to use weaker passwords. One reviewer wrote:

> "Not only do I have too many passwords now to remember, but I just know reusing the same password for multiple sites is a big no-no, even if they are really good passwords! This solves both issues."

Another wrote:

> "I use this almost every day. Having everything in one spot has made my life much easier. Without my passwords in the computer, I feel they are much more secure."

Risks are Negligible or Could Be Mitigated. Some reviewers recognized that there are risks associated with using password logbooks. However, they felt these risks were insignificant. For example, one reviewer wrote:

> "Yes, obviously, if your book gets stolen that's a problem but it's a problem if your password app account is compromised or someone reads your thoughts, too, so everything is a risk and I will take a risk for convenience."

Others believed that naive usage might be risky, but taking appropriate precautions would mitigate these risks. For example, one reviewer wrote:

> "Okay. I have read the objections to this means of keeping one's passwords–and I get it–but there need not be any problems! I would not travel with this anyway, so that risk is eliminated. Still, there are ways to enter the info into this book that make it impossible for anyone to sabotage you, by stealing your info. I do not write out the full names of the websites I frequent; I find creative ways to abbreviate the names, so that no one

> other than myself could guess what the site is. I select passwords/phrases that I will still know, even after I substitute x's or underscores (_) for some of the characters. So, again, unless someone is psychic, they will not be able to get my pass codes. There is plenty of room to write–perhaps, too much, as my only complaint about this book is that it is too big. I would have preferred one no larger than a 3 x 4, but decided to go with it, given all the other positive reviews. Size makes it easily hide-able enough in your home. Use your common sense and this will be just fine. :)"

User Perceptions of Risks of Using Password Managers. Almost all reviewers valued the password logbooks they purchased, but there were a small fraction of reviewers who were dissatisfied with their purchase, and a minuscule fraction who disapproved of password logbooks altogether. Given the subpopulation we're considering of reviewers who had purchased password logbooks on Amazon, this skew makes sense. One reviewer said they purchased and sent a password logbook as a gag gift to a friend who works in security. The reviewer then cautioned against using password logbooks, suggesting password managers as a more secure alternative. As stated above, this reviewer was an anomaly amongst verified purchasers.

To explore this theme further and to see how other users would respond, we temporarily broadened the scope of our reviews to include a small set of unverified purchase reviews, in which we saw more criticism of password logbooks. Many reviewers suggested password managers to be a lower-risk solution. Some justified their statements. Some rebuked users for using password logbooks. These reviews led to interesting and surprising dialogue that shed light on why some users choose to use password logbooks even when they're aware of password managers. For example, one individual no longer trusted their password manager because the antivirus software they were using classified it as a trojan. Another stated:

> "I purchased LastPass a year ago and was dismayed to get an alert from them that their system had been compromised. My data wasn't compromised, but decided then and there nothing is really safe. I prefer to have something that I have control of, like this small book, than give my information over to a service where I have no control of where information is stored or how it is protected."

Curiously, one individual stated that the book was a bad idea, but then suggested a method to generate what they deemed strong, memorable passwords; however, the suggested method is easily susceptible to a password reuse attack by an adversary who notices the pattern.

2.5.11. Tricks and Advice

Reviewers were very willing to share what they thought of as clever tricks and prudent advice. This included writing down passwords in pencil so they could easily be erased, writing in what is effectively a password hint in lieu of the actual password, storing the password logbook securely, leaving out contact information so an adversary cannot identify who the book belongs to (though some purchasers appreciated space for listing contact information), etc.

For one example, a reviewer wrote:

> "Write your entries in pencil! Even if it is for an account that you suspect will always have the same information, there are plenty of reasons that entries in pencil are beneficial. Your account could be hacked forcing you to change a password, you could change banks or email accounts, the website address to the business may even change. Much simpler and cost effective to erase/edit an entry, rather than buy a new book."

Another reviewer wrote:

> "I sometimes consult with people who have problems remembering their passwords. First, I teach them a 'reminder' method, then I gift to them this little book, where they write their password 'reminders.' Using a 'reminder' method (where you don't put the actual password, but instead something that reminds you of the password), this book is invaluable. And if it goes missing, it's not the end of the world because nobody will understand how to use it. And, if you're smart, you won't put your name (or any other identifying info) in the book. This should not be the ONLY place you have passwords, because that would be like not backing up at all, and we've all heard those horror stories. But for quick reference at home or in the office, it's a great idea. Like the 'little black address book,' it's indispensable."

Yet another wrote:

> "To make your passwords in your book even more secure, add an extra special character that you never use in any password. Then ignore that special character whenever you enter your password. For example, put @ into each password just as a ruse. Or use some variation, such as ignoring the eighth character in each password."

Section 2.6

Discussion

We acknowledge the irony of users writing down account credentials in password logbooks, some of which are even labeled "Password Logbook," violating the often prescribed security advice that you should never write down your passwords. Adoption of these books is at least partially rooted in well-intentioned, but potentially counterproductive, password policies and password authentication protocols. The cognitive burden of having to remember associations between service names, usernames, and passwords, along with other challenges such as having to remember answers to security questions or remembering a password to an old account after a mandatory password reset, leads users to use these password logbooks. For example, one reviewer wrote:

> “My memory is not bad but every website now wants passwords and security questions. I am so tired to trying to remember every one.”

That is, we may be seeing an *uncanny descent*: increasing the complexity of password policies with the expectation of improving security may actually make things worse by driving users to engage in riskier practices, such as writing down and reusing passwords, to alleviate the increased cognitive burden of managing their passwords under the new password policies.

There’s also the reverse irony that these password logbooks may provide better security than the alternative password management strategies users employ. The knee-jerk reaction of discrediting the use of password logbooks as an unacceptable form of circumvention that only worsens security may be premature or not sufficiently nuanced to reflect the reality of regular users’ lives. Password logbooks often supplant other, more risky forms of circumvention and alternative password management strategies. Moreover, many users don’t believe they’re capable of memorizing many strong, unique passwords which is why they turn to password logbooks; many view password logbooks as a convenient tool that provides more convenience, better security, and/or better organization than their current password management strategy. For example, one user wrote:

> “I also like that I don’t have to use the same password for every site because it’s all I can remember.”

A number of reviewers expressed awareness of the risks associated with using password logbooks, but used them despite the risks. The reviews suggest that many users employ a rational decision-making process to settle on password logbooks; they determine alternatives provide less value in terms of convenience or security and, in many cases, both. This is in agreement with the literature, e.g., [78].

While we make no claim that password logbooks are optimal or even good options for password management, we are suggesting, as has been suggested by reviewers, that in the absence of password logbooks, some users would be at greater risk. Prescribing good security behavior that users don't adopt may be worse than giving users suboptimal, but still beneficial, advice that they actually follow. That is, we should not expect users to engage in the most secure behaviors, but we should instead nudge users toward the best security solution amongst those they're willing to put up with. We must also acknowledge the limitations of proposed security solutions. For example, while some password managers may be more secure than password logbooks in general, in the event that the user is unexpectedly incapacitated, they may not provide a mechanism to transfer account credentials to family. Indeed, earlier we quoted a reviewer who used a password manager, but still had a backup password logbook for this very reason.

Section 2.7 Method

To conduct our analysis, we downloaded both the product pages for each password logbook, as well as Amazon Verified Purchase reviews for them.

We searched amazon.com for the key phrase "password logbook." We then constrained the search to include only those products classified under the category of "Books." From the results, we obtained a list of 132 password logbooks in sorted order of reviews with at least one review, with the most reviewed book appearing first. However, five of these did not adhere to our descriptive definition of a password logbook, narrowing our dataset to 127 books.[4].

We downloaded both the product pages for the 127 password logbooks, as well as all 4,778 Amazon Verified Purchase reviews for them. Next, we removed duplicate reviews; if we found two or more reviews that had the same author, review title, and review text, we kept only one copy of the review. Duplicate reviews appeared for various reasons. Some password logbooks were listed as different products but were just a different edition of another book, which had the exact same set of reviews; 7 of the 127 password logbooks were doubly-listed, accounting for 413 duplicate reviews. 35 more duplicate reviews were found using a script. Additionally, 4 more password logbooks were removed from our dataset: 1 logbook was removed because it only had a single review, which was a duplicate; the 3 other logbooks were removed because they only had non-Verified Purchase reviews. These steps and manual inspection of reviews reveal most duplicate reviews were attributable to doubly listed logbooks and instances of reviewers buying multiple logbooks and leaving the same review for all of them, e.g., because the review involves a comparison of them or because the reviewer bought multiple editions as gifts and left the same review for each edition as they are essentially the same other than cosmetic differences. That said, we did see a few fraudulent reviews. Please see Section 2.8 for further details.

Our final dataset comprised 116 password logbooks and 4,330 reviews for these password logbooks, with duplicates removed. We extracted relevant data from the reviews and two of us applied grounded theory to determine common themes from the reviews.

[4] The five discarded books roughly fell under two categories: regular address books, e.g., `https://www.amazon.com/dp/1593593899`, and books that served as guides to organize one's records with space to record things like tax records, property records, and even passwords, e.g., `https://www.amazon.com/dp/1413323154/`.

Section 2.8

Limitations and Advantages

The study has the following limitations:

- **Some Reviews May Be Fake or Biased:** Any study on a corpus of Amazon reviews may suffer from the presence of illegitimate reviews. For some examples, the reviewer may have bought the product at a discounted rate in exchange for leaving a review; the reviewer may have been paid to leave a positive review; the reviewer may even have been paid to leave a negative review for a competitor product; the reviewer may have left a review to gain credibility. To address this problem, we restricted the data set to comprise solely Amazon Verified Purchase reviews for products. However, we still came across some reviews that we believe to be fake, although we believe they constitute only a small fraction of all reviews. Moreover, the primary motivation behind this study is to glean insights into how some users think about password authentication; the existence of a few fake reviews has negligible impact on this pursuit.

- **The Sample Set:** Any study on a corpus of Amazon reviews also inherently limits its sample set to authors of Amazon reviews. In our case, this meant that reviewers—aside from a few reviewers, e.g., one reviewer purchased a password logbook as a gag gift for a friend—were drawn from the subpopulation of general users who willfully circumvented recommended security practices, bought a password logbook on Amazon, and wrote a review for said password logbook. While this sample undoubtedly does not reflect the entire population of computer users, we believe there's valuable information to be had in these reviews—and, indeed, the sample reveals the existence of a subpopulation who engages in the practice of using password logbooks.

Despite these limitations, our approach—and ones similar to it—have a number of benefits:

- **The Sample Set:** Our approach is less susceptible to some other selection biases common to other studies. For example, many academic studies involve a disproportionately large fraction of college students. Some themes we saw simply would not emerge with such a sample set. For this reason, our findings in this study nicely complement those in the existing literature.

- **Scale:** Our data set of reviews comprises 4,330 reviews and has no monetary cost. In general, approaches like this—ones that looks at user product reviews, posts in forums, comments on articles, and so forth—provide great scale for minimal cost.

- **Reviews are Volunteered:** Perhaps the strongest aspect of this approach is that the information contained in these reviews is provided to us directly from the user without any request for information. A number of the biases present in face-to-face interactions, surveys, or other solicited feedback, is not present here. Moreover, we speculate that the reviewer's state of mind is different in writing these reviews than it would be if their feedback were solicited, regardless of such biases. That is, the user isn't primed to deliberate about their motivations, beliefs, and behaviors regarding passwords, as they likely would be in a survey.

Section 2.9

Future Work

While this work provides valuable qualitative data about certain users, due to the limitations mentioned in Section 2.8, we cannot provide meaningful quantitative data about users in general. Follow-up studies, such as surveys and behavioral experiments conducted on a representative sample of a broader subpopulation of users that further explore the themes we mentioned may provide valuable quantitative data to further assist in suggesting security policies and mechanisms to employ and to suggest how we should communicate with and advise users regarding security.

Similar approaches to this, that involve analyzing other reviews, forum posts, and comments on articles, may serve useful in developing a better and understanding of the user. Data sources like Amazon customer reviews also provide valuable metadata. For example, review dates may enable researchers to study how user perceptions and attitudes change over time, which is hard to attain retroactively via other means. Similarly, comparisons between reviews on, say, amazon.com and amazon.co.uk, would enable researchers to study regional variations in beliefs and behaviors. It would also be enlightening to explore data sources that provide dialogue amongst users.

Section 2.10 Conclusion

We examined a subset of available password logbooks on Amazon and their reviews. The sheer existence and diversity amongst password logbooks and the magnitude of reviews available for them was illuminating in its own right. Moreover, a number of interesting themes emerged in the process of analyzing reviews, some of which provide new insights into user beliefs and behaviors. Reviewers felt the risks of using password logbooks were negligible and could be mitigated by taking appropriate steps. A couple of respondents had used password managers but resorted to password logbooks after a bad experience—perhaps the most interesting finding was from a reviewer who stated their antivirus software considered their password manager a trojan, leading them to find a new solution—or due to a lack of a critical feature, e.g., sharing passwords with family members. People also generally felt that the password logbooks significantly improved their digital security by eliminating worse password management strategies, such as reusing passwords across services. Overall, we feel the approach of harnessing freely available user data to gather insights about users is effective.

Chapter 3

Measuring the Security Impacts of Password Policies Using Cognitive Behavioral Agent-Based Modeling

Agent-based modeling can serve as a valuable asset to security personnel who wish to better understand the security landscape within their organization, especially as it relates to user behavior and circumvention. In this chapter, we argue in favor of cognitive behavioral agent-based modeling for usable security and report on our work on developing an agent-based model for a password management scenario. We explain the password management simulation, conduct a sensitivity analysis, and discuss security implications, e.g., an organization that wishes to suppress one form of circumvention may benefit from endorsing another form of circumvention. These sorts of simulations are particularly valuable in averting what we call *uncanny descents*, instances where turning up the security dials in the hope of improving aggregate security actually has the opposite effect. In other words, agent-based simulation is a useful tool in recognizing and addressing intent-outcome mismatches.

Section 3.1

Introduction

Agent-based models incorporating user behavior, emotion, and cognition can serve as valuable tools that assist computer security personnel design, implement, and maintain security systems, devise security policies, and employ security practices that are congruent with security and other organizational objectives.

Indeed, as the current state of security practice indicates, we need these sorts of tools. Our interviews, surveys, and observations reflect many examples where security solutions fails to accommodate users. Such mismatches between user needs and security policies and mechanisms often induce circumvention, thereby undermining overall objectives. Even if one could design adequate security policies and mechanisms *a priori*, the dynamic nature of software systems, user needs, and organizational and environmental changes would necessitate frequent readjustments. Consequently, we need tools that allow us to better understand this complex security landscape. Such tools can help us to evaluate the costs associated with security solutions, identify unintended side effects of said solutions, and pinpoint usability issues that drive user circumvention, among other things.

dash [31, 30], an agent-based simulation framework that supports the dual-process model of cognition, reactive planning, modeling of human deficiencies (e.g., fatigue, frustration), and multi-agent interactions, enables us to create such tools. In dash, users are represented as agents with weighted goals, plans to achieve those goals, attributes, knowledge, and abilities. These agents use mental models and have perceptions of the world that often depart from reality. Agents, in accordance with their mental models, take actions, observe and interpret events, and communicate. They dynamically compute and recompute goals and the plans they use to achieve them. DASH models may better enable security personnel to (a) identify weaknesses in security policies and mechanisms, e.g., workflow impediments that prompt user circumvention, (b) estimate the likelihood of user engagement in workarounds, (c) gauge the number of inescapable security infractions from policy-workflow mismatches, (d) estimate the values of security and organizational objective functions, (e) test the accuracy of proxy security measures, and (f) measure how shifts in the environment affect security. A cognitive and behavioral approach to modeling can provide insights into the effectiveness of informing users of practical needs for security, implementing a feedback loop, imposing more stringent policies or harsher penalties for circumvention, and more.

Agent-based modeling is particularly useful in scenarios where security in practice radically differs from security in the abstract, where it's extraordinarily challenging to anticipate how emotions, cognitive biases, and other human deficiencies may affect user behavior. Indeed, in order to get security right it is critical that we understand how users interact with our systems. And we must adapt our systems to our users—not expect our users to adapt to our systems—so as to induce "good" behavior. [32, 4] In previous work [100], we discussed the potential for agent-based models to be applied to predict human circumvention of security, relayed an anecdote regarding timeouts in a medical setting, explained preliminary work, and discussed our future directions for building such models. The work presented in this chapter follows up on that work by detailing our progress on modeling a password management scenario.

The password management scenario involves establishing password polices for an enterprise. In theory, having a policy that requires users to use strong passwords, to never write them down, and to never reuse them across sites would improve security. In practice, users commonly circumvent password policies due to perceived cognitive limitations, fatigue, frustration, and work culture. Moreover, password choices and password management practices for one service may affect the choices and practices for another, making the services interdependent. By applying agent-based models, security personnel can better understand this complex environment, estimate measures of aggregate security that incorporate circumventions, risks, and costs, and ultimately make better decisions.

This chapter is structured as follows. In Section 3.2, we introduce the DASH modeling framework. In Section 3.3 we investigate the password modeling scenario, detail our DASH modeling work, perform a sensitivity analysis, and discuss results and takeaways. In Section 3.4 we discuss future work including the autologout scenario. In Section 3.5 we conclude.

Section 3.2
The DASH agent modeling platform

The DASH agent modeling platform provides a framework and a set of capabilities for modeling human behavior [31], designed to capture observations from human-centered security experiments, e.g., [52]. In order to model human task-oriented behavior, which is both goal-directed and responsive to changes in the environment, DASH includes a reactive planning framework that reassesses goal weights and plans after receiving input after an action [37]. In order to model deliberative behavior, DASH includes an implementation of mental models following the approach of Johnson-Laird and others [89] and a simple framework for evaluating costs and benefits of alternative worlds. This approach adopts the view that users follow essentially rational behavior when making decisions about online actions including security, but typically have an incomplete or incorrect model of the security landscape.

In order to model bounded attention that affects human decision-making, particularly under stress or cognitive load, DASH adapts psychology's *dual-process framework*

[182] in which two modules provide alternative suggestions for the agent's next action. The first is a deliberative system that uses the mechanisms for planning and mental models to arrive at a decision, and the second is a stimulus-driven system that matches surface properties of a situation to find an answer. Once an agent has experience in a domain, the stimulus system provides good answers most of the time while an inexperienced agent may need to fall back on deliberative reasoning more often. Under stress, time pressure, or cognitive load, the deliberative system may not complete, or the stimulus system may gain increased weight, leading to impulsive behavior that may not be correct.

Other cognitive architectures such as SOAR [109] or ACT-R [13] provide many of the same behaviors. One distinguishing factor of dash is that its stimulus system is not related to the deliberative system by a compilation learning process and can often produce results that differ qualitatively rather than in terms of speed. dash also provides support for mental models and tradeoff analysis as more fundamental components.

Section 3.3 The Password Management Scenario: Security Dependencies Introduced by Workarounds

With an understanding of dash we now discuss our password simulation. In Section 3.3.1 we cover preliminaries including a discussion of password problems and related work. In Section 3.3.2 we explain how our simulation works. In Section 3.3.3 we present the parameters used in our simulation and the sensitivity analysis. In Section 3.3.4 we discuss how we conducted the sensitivity analysis. In Section 3.3.5 we enumerate sources of error. In Section 3.3.6 we present our results. In Section 3.3.7 we provide takeaways.

3.3.1. Preliminaries

In terms of usability and security, many consider passwords a failure. Users are notorious for choosing weak passwords. In an effort to mitigate the security risks linked to weak passwords, many services now require users to choose passwords that satisfy complex password composition rules. Unfortunately, this brings with it a slew of other security challenges, e.g., [32, 4, 63, 65]. Users who are unable to cope with the increased cognitive demands of having to remember dozens of passwords resort to circumventing password policies and employing poor password management strategies; they write passwords down on Post-it notes, reuse passwords across multiple services with little or no variation, and leave passwords in plaintext files on their computers. However, perceived cognitive limitations are not the only impetus for user circumvention of password policies. In some domains, users need to share information with others who have different access rights than themselves, but the “proper” channel for information sharing is slow and inefficient. So, they share passwords instead [32].

Services are culpable too. Some services effectively discourage strong passwords by setting low ceilings on password length, disallowing special characters, using easily guessable security questions, and assigning default passwords that are often left unchanged. Others impose excessive password complexity requirements and require frequent password resets, which further incentivizes users to circumvent. In recent years, many services have also been the target of massive password breaches; in some cases, they have even exposed passwords to malicious actors in cleartext. Moreover, due to password reuse, risks associated with poor password practices are not confined to those services that are lax about password security. That is, even if a service makes a legitimate effort to secure their users’ passwords, those passwords could be compromised by vulnerabilities at other services [87].

While tremendous effort has been spent on trying to replace passwords, it has been met with questionable success. Bonneau et al [35] compared passwords to other authentication schemes in three domains: usability, deployability, and security. They showed that no alternative authentication scheme dominates passwords.

In short, passwords pose numerous memorability and usability challenges that frequently manifest in user circumvention. They pose confidentiality, risk mitigation, and public perception challenges for services. And they do not appear to be going away any time soon. This motivates the need for better techniques to both assess the costs and mitigate the risks associated with password policies.

Numerous recent studies have looked into password modeling. Shay et al. [171] developed a simulation to examine the effectiveness of password policies. Choong [45] proposed a user-centric cognitive behavioral framework for the password management lifecycle, from password creation to password reset. SimPass is a highly configurable agent-based model for measuring the efficacy of password policies [161]. Our work is similar to SimPass in that we've developed a password simulation with knobs that can be adjusted to measure aggregate security associated with password policies under different circumstances. Whereas SimPass employs numerous parameters to better understand password management scenarios with minimal assumptions, we adopt the view that many of these parameters cannot be known, nor do they need to be known, *a priori* to have a useful predictive model. Our simulation instead relies on a smaller number of parameters with more underlying models, especially those related to cognition and behavior. For example, there are underlying models for a password belief system and cognitive burden. While this approach provides valuable insights into the cognitive and behavioral factors that affect security, it necessitates different kinds of modeling assumptions.

3.3.2. Simulation Details

Our simulation models human users interacting with computer systems that employ username and password authentication. Specifically, agents construct plans to achieve high-level subgoals for creating accounts, signing in to accounts, and signing out of accounts. Each subgoal is broken down by the agent into a series of steps during action invocation as determined by the agent's beliefs, the agent's cognitive burden, and other factors. To better understand how these processes work, we first explore the underlying models for the agent's belief system and the agent's cognitive burden.

Let us first briefly discuss the agent's password belief system. For this discussion, we limit ourselves to passwords; a similar model exists for usernames. For service S and password P, let $V_{S,P}$ denote the strength of the agent's belief that password P is the correct password for their account on service S. During the sign-in process, these password strength values are used to determine whether the agent recalls a password for a given service and, if so, which password the agent recalls. Agents slowly forget passwords during periods of non-use as reflected by reductions in password strengths. In fact, following every user action, all password strengths over all services are decremented by service-specific password forget rates.

We now discuss the underlying model for cognitive burden. As before, we limit our discussion to passwords. The model uses a generalization of the Levenshtein distance to sets and makes use of an openly available Prolog implementation of Levenshtein distance [46]. The Levenshtein distance between a string S_1 and S_2, denoted as $Lev(S_1, S_2)$, is the minimum number of character insertions, deletions, and substitutions required to convert S_1 into S_2. For set S, define the Levenshtein measure $L(S)$ as the weight of a minimum spanning tree T over the vertex set $S \cup \{\epsilon\}$ for which edge weights are specified as $w(v_1, v_2) = Lev(v_1, v_2)$. Here, ϵ denotes the empty string. The cognitive burden of a set S_P of passwords in our simulation is approximately $L(S_P)$.

There is also a small cost associated with mapping passwords to services in memory. We very roughly approximate this by including an additive factor of 1 for each service that has a corresponding password that is in the agent's memory.

Equipped with an understanding of these two underlying models, we can now look more deeply at the subgoals associated with creating an account for a service and signing in to a service.

During account creation, the agent must first construct a username and password combination. If the agent's cognitive burden is under a specified threshold, the password reuse threshold, the agent chooses the weakest password they can think of that satisfies the password composition requirements. If, however, the agent's cognitive burden exceeds this password reuse threshold, the agent attempts to recycle an existing password before considering a new, unique password. The particular password chosen for reuse is determined by the password reuse priority parameter which specifies whether the agent should reuse the longest or shortest viable password. Once an account has been created, the agent may opt to either memorize their password or write it down. This process is again determined by comparing the agent's cognitive burden to a specified threshold, the password write threshold. If the agent's cognitive burden is under the threshold, the agent will try to memorize the password; else, the agent will write it down. If the agent opts to memorize password P for service S then the password strength $V_{S,P}$ will be initialized to 1, while if the agent instead opts to write down the password, $V_{S,P}$ will be initialized to 0.5. And, in both cases, all S-specific password strengths associated with passwords different than the chosen password are set to 0; that is, $V_{S,P'}$ will be set to 0 for $P' \neq P$. Additionally, during account creation the service-specific password forget rate is initialized to a model parameter entitled initial password forget rate. While we've discussed the process of account creation, the same processes largely apply to the password reset process, the primary difference being that the agent will not create a new username.

When an agent wishes to sign in to service S, they first attempt to recall their password for the service. This is done by choosing the password P with greatest S-specific password strength. If $V_{S,P}$ exceeds a parameter called the recall threshold, the agent attempts to sign in using P. If the agent cannot recall a password, that is, if there is no password with S-specific password strength that exceeds the recall threshold, the agent checks to see if they wrote down a password. If the agent did write down a password, then the agent uses that password; else, the agent resets the password.

As discussed earlier, after each action is performed, password strengths are decremented by service-specific password forget rates. These forget rates are initialized to an initial password forget rate during account creation and password resets, and they are changed during sign-in attempts. Whenever the agent enters a password P for a service S and it is accepted, the password forget rate for that service is halved, the password strength $V_{S,P}$ is set to 1, and, for all $S' \neq S$, $V_{S',P}$ is strengthened by the product of the password forget rate for S' and the strengthen scalar, a model parameter. When the agent enters a password P for a service S and it is rejected, $V_{S,P}$ is set to 0. While this model is not faithful to reality (e.g., it does not incorporate the time duration between successive recalls) we, again, believe it serves as a good, simple first approximation.

To assess the risk of password compromise, we consider three attack vectors. The first is a direct attack in which the attacker either exploits a service vulnerability or brute forces the password. This is a function of a direct attack risk scalar and a raw password strength function that maps passwords to strength values. The second is an attack wherein the attacker sees the agent's password written down and uses it to access the agent's account. If the password has been written down the risk for this attack is equal to a model parameter that specifies the stolen password attack risk; else, it is 0. The third attack is an indirect attack in which the attacker, using one of the previously mentioned attacks, discovers the agent's password for another site, and then reuses that password to sign in to the agent's account for the target service. The risk of this attack is equal to one minus the probability of being safe from indirect attacks, where the probability of being safe from indirect attacks is the product of probabilities of being safe from indirect attacks for each individual service. The probability that a service S is safe from an indirect attack stemming from S' is the product of a model parameter, the reuse attack risk, and the probability that S' is not compromised by one of the two aforementioned attacks. For future discussion, we define the security measure M to be the probability that a service is safe from attacks, averaged over all services.

Services are loosely grouped into four classes based on the complexity of their password composition policies: weak, average, good, and strong. Complexity requirements affect the minimum length, minimum number of lower-case alphabetic characters, minimum number of upper-case alphabetic characters, minimum number of digits, and minimum number of special characters required for a password to be accepted. All member services of a single class use the same process to generate their password composition policies.

We now briefly explain the code and primary processes in the simulation. The simulation involves agent-side code that is responsible for choosing and performing agent actions and a world hub that is responsible for carrying out all service processes, keeping world state, and printing statistics. A target service is also passed to the world hub. Printed statistics include the number of accounts that have been created, the number of usernames and passwords each agent has written down, the number of usernames and passwords each agent has memorized, the number of passwords resets each agent has performed, and aggregate security measures M associated with each agent's set of accounts. Additionally, for each agent, the hub prints similar statistics for the target service.

3.3.3. The Parameters

For the purposes of better understanding our model and gleaning insights into the security implications of different password policy settings, we performed a variation of one-factor-at-a-time sensitivity analysis. Although many parameters are highly interactive, this approach still provides valuable insights. Here, we review the parameters and state the fixed values used for analysis. In the subsections immediately following this one, we will discuss the method, comment on sources of error, explain our results, and state key takeaways.

Below, we specify the fixed value we use for each parameter considered in our sensitivity analysis. We also provide a short description of the parameters.

Initial Password Forget Rate:. 0.0025

This parameter specifies the initial password forget rate that is set for a service during account creation and password reset.

Recall Strengthen Scalar:. 4

This parameter affects the amount that a password belief is strengthened for one service when the password under consideration is successfully used for another service. Specifically, when an agent successfully signs in to service S with password P, for each $S' \neq S$, the password strength value $V_{S',P}$ is incremented by the product of the recall strengthen scalar and the password forget rate for S'.

Recall Threshold:. 0.5

This parameter specifies the threshold over which the agent can recall passwords.

When the agent is trying to sign in to a service S, the agent will consider the password P with highest strength value, $V_{S,P}$ associated with that service. If $V_{S,P}$ exceeds the recall threshold, the agent will attempt to sign in with password P. Else, the agent will be unable to recall a password and will instead resort to another action.

Password Reuse Priority:. long

This parameter can take on one of two values: short or long. When an agent attempts to reuse an existing password during account creation or password reset for a service, this parameter specifies whether the agent reuses the shortest or the longest password that satisfies the password composition requirements for the service should there exist a recallable password satisfying the password composition requirements.

Password Reuse Threshold:. 40

If an agent creates a new account for a service or resets their password for a service and the agent's cognitive burden exceeds the value of this parameter, the agent will opt to reuse an existing password.

Password Write Threshold:. 60

If an agent's cognitive burden exceeds the value of this parameter after creating an account for a given service or resetting their password for a service, the agent will opt to write down the password instead of attempting to memorize it.

Direct Attack Risk:. 0.25

This parameter affects the probability that an account may be compromised directly via a service vulnerability or brute force attack, not a stolen password or reuse attack. It effectively acts as a scalar for the password strength associated with a given service to determine the direct attack risk.

Stolen Password Risk:. 0.25

This parameter specifies the probability that the attacker may find the agent's password written down and successfully use it in an attack.

Reuse Attack Risk:. 0.25

This parameter specifies the probability that an attacker successfully launches a reuse attack on a service S by exploiting a given direct attack or stolen password attack on another service S'.

Distribution of Services:. (6,6,6,6)

This parameter is a vector of four integers that specifies the distribution of services according to the strengths of their password composition policies. The shorthand (W, A, G, S) means that W, A, G, and S services employ weak, average, good, and strong password composition policies respectively.

3.3.4. Method

We performed a one-factor-at-a-time sensitivity analysis wherein we decided *a priori* on fixed values for each of the ten parameters specified in Section 3.3.3. We varied each parameter within a constrained, feasible parameter space and recorded the aggregate security M (refer to Section 3.3.2 for more details on M) for six independent trials for each parameter configuration we considered. Our sensitivity analysis is actually a slight variation of the traditional one-factor-at-a-time approach in that, for the distribution of password composition policies parameter, we performed a series of trials for three different configurations of the cognitive thresholds (i.e., password write threshold and password reuse threshold) to better understand the interplay between the three parameters.

For all but one parameter, we stopped simulations when the agent's minimum per-service password forget rate dropped below 0.0005. The exception occurred during testing of the initial password forget rate parameter, which was performed first. While testing the password forget rate parameter, we stopped simulations when the minimum per-service password forget rate dropped below 0.00025.

After gathering data as described above, we generated plots with error bars corresponding to the standard deviation.

3.3.5. Sources of Error

Computer-based arithmetic accounts for one source of error. While not a true source of error, we do see some peculiarities in our graphs due to the use of a finite set of approximately thirty passwords and the use of a step function for evaluating password strength. Though the password list and password strength evaluation function are in some sense parameters, specifying a feasible solution parameter space for them and varying them accordingly is beyond the scope of this work. Last, we recognize that performing only six trials for each configuration of parameters is a limitation.

3.3.6. Results & Analysis

Here, we present the results and analysis.

Initial Password Forget Rate. In Figure 3.1 we see that increasing the initial password forget rate reduces security. Our belief is that as we increase the initial password forget rate users are more inclined to reset their passwords and write down the newly reset passwords during the process.

Recall Strengthen Scalar. In Figure 3.2 there seems to be a slight increase in security as we increase the recall strengthen scalar. While this may just be error, this may also be in part due to a reduction in passwords being written down as the value of this parameter increases.

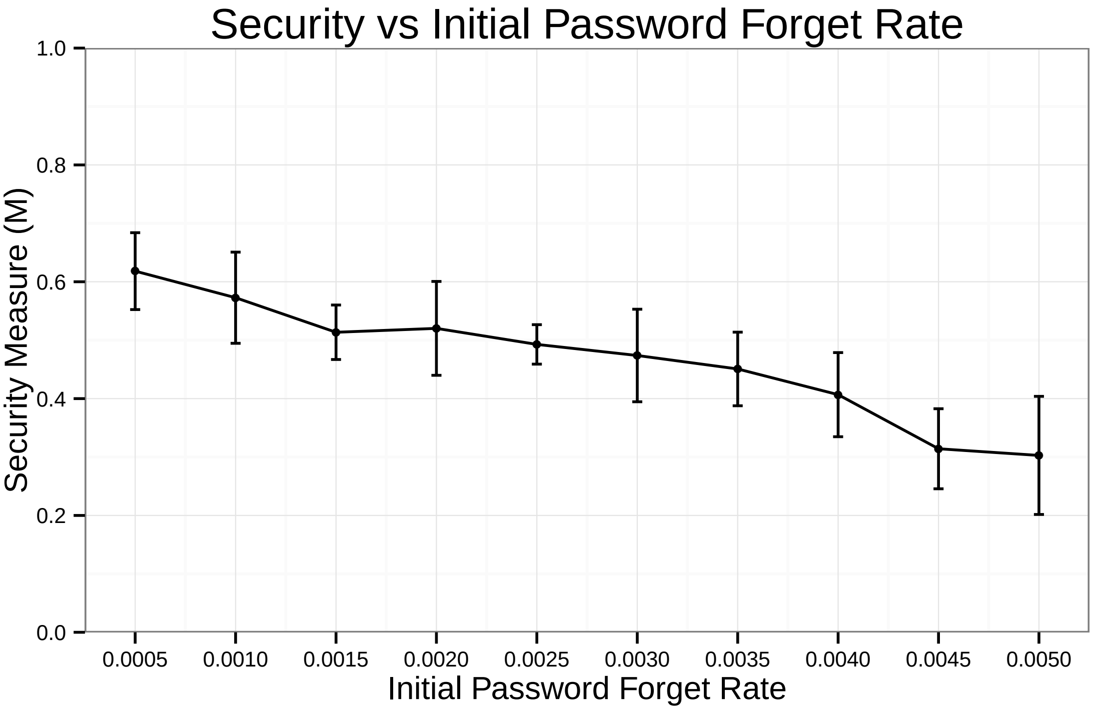


Figure 3.1: Security vs. initial password forget rate.

Recall Threshold. In Figure 3.3 we see that increasing the recall threshold decreases security. This indeed makes sense. A higher recall threshold means it is more difficult for the user to recall passwords. So, the user will frequently reset their passwords instead of remembering them. After the user has accrued a large number of accounts, these frequent resets will lead the user to circumvent as a coping mechanism.

Password Reuse Priority. We found that having agents reuse the shortest acceptable password leads to a higher security measure than reusing the longest password. With a short password reuse priority we saw a mean security measure of $M = 0.5222$ with a standard deviation of 0.0616. With a long password reuse priority we saw a mean security measure of $M = 0.4528$ with a standard deviation of 0.0770. One possible explanation for this discrepancy is that a tendency toward

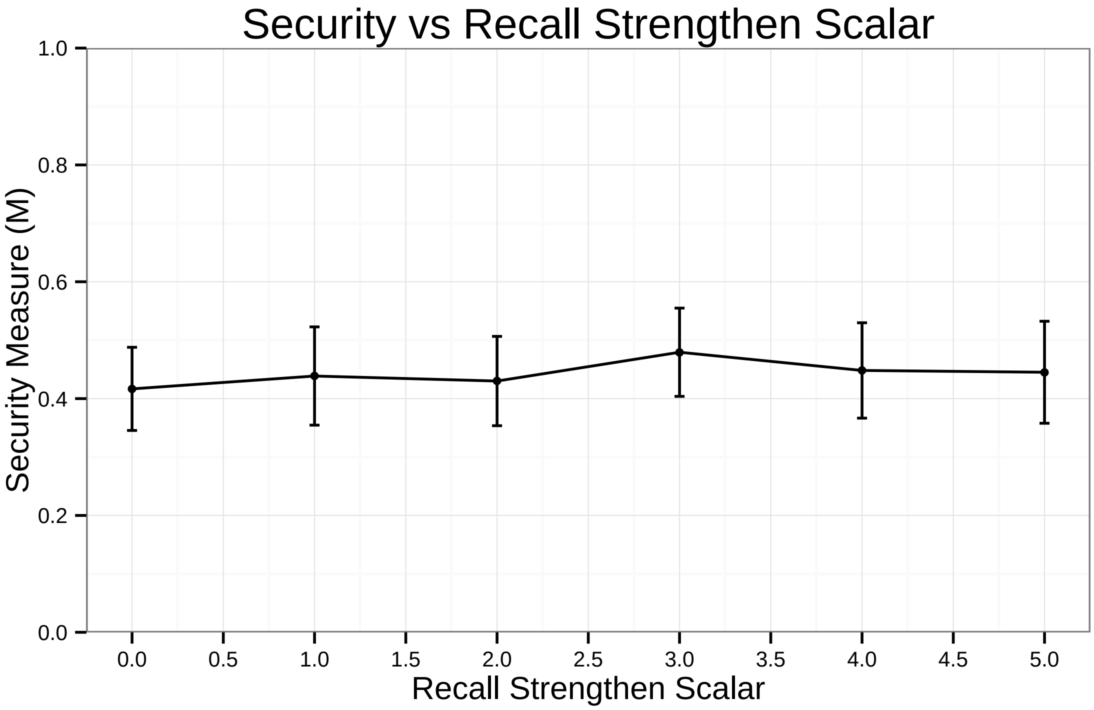


Figure 3.2: Security vs. recall strengthen scalar.

reusing shorter passwords reduces the likelihood that a single password is reused across most accounts. That is, since password composition policies vary across services, longer passwords will be more likely to satisfy a greater fraction of password composition policies than shorter ones. Ergo, longer passwords will be more susceptible to reuse attacks.

Password Reuse Threshold. As expected, in Figure 3.4, increasing the password reuse threshold improves security.

Password Write Threshold. At first glance, Figure 3.5 may seem a bit surprising. When the password write threshold is very low, M is reasonably high. As we increase the password write threshold, we see a dip in M. And, as we further increase it we see M rise to a value slightly above its value when the password write threshold was

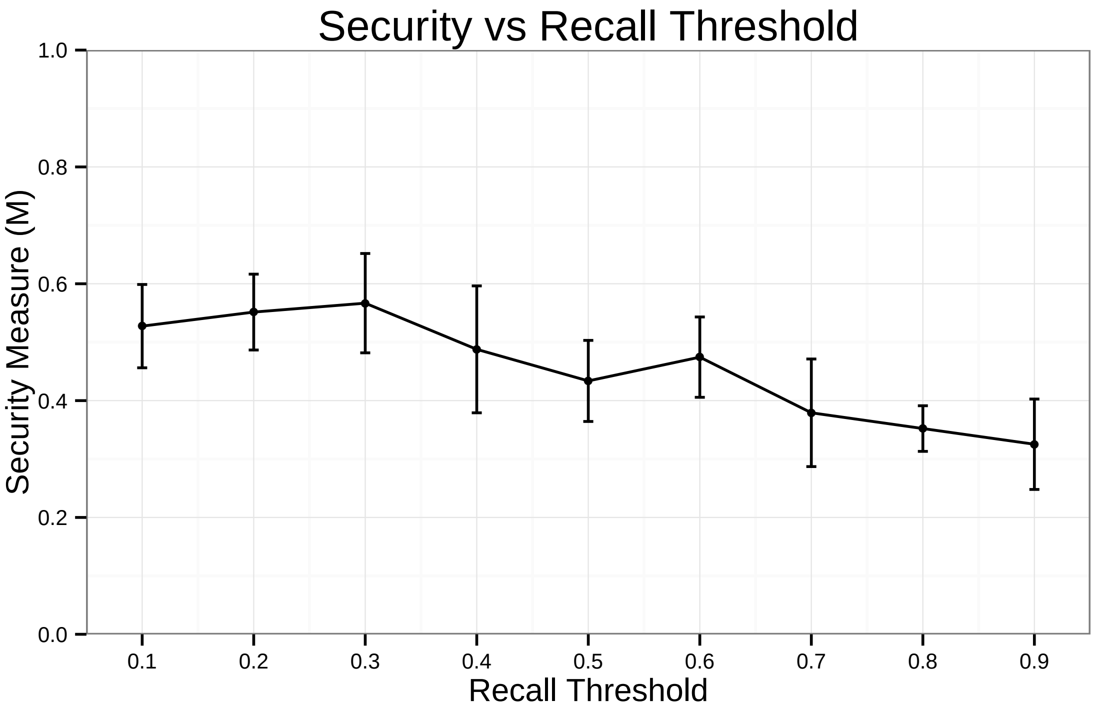


Figure 3.3: Security vs. recall threshold.

very low.

Our rationale for this behavior is as follows. When the password write threshold is low, under 40, users do indeed write down passwords, but writing these passwords down means that the passwords contribute less to the cognitive load of password remembrance; this leads to a larger set of unique passwords at the cost of more passwords being written down, which is a net win as determined by the parameter settings of direct attack risk, reuse attack risk, and stolen password risk that determines M. We see a dip when setting the threshold between 40 and 80 because while users are less inclined to write passwords down during this range, they will be more inclined to reuse passwords as passwords that are not written down contribute a larger cognitive burden. For thresholds over 80, users may still reuse more passwords, but the gains from not writing down passwords finally begins to outweigh gains from not reusing

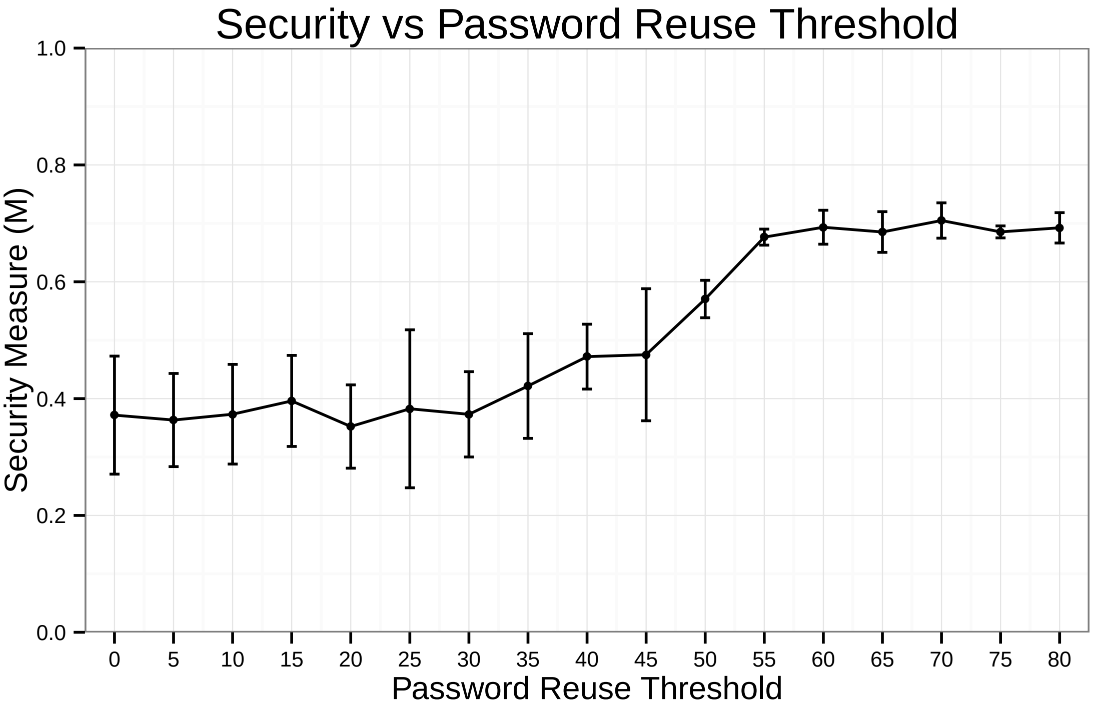


Figure 3.4: Security vs. password reuse threshold.

passwords.

Direct Attack Risk. In Figure 3.6 we see that increasing the direct attack risk value reduces security as expected.

Stolen Password Risk. In Figure 3.7 we see that increasing the stolen password risk value reduces security roughly as expected. We do see an unusual local maximum for a stolen password risk value of 0.5. We attribute this solely to error because we performed too few trials.

Reuse Attack Risk. In Figure 3.8 we see that increasing the reuse attack risk value reduces security as expected. We see a peak at 0.625, but we again attribute this to error due to an insufficient number of trials.

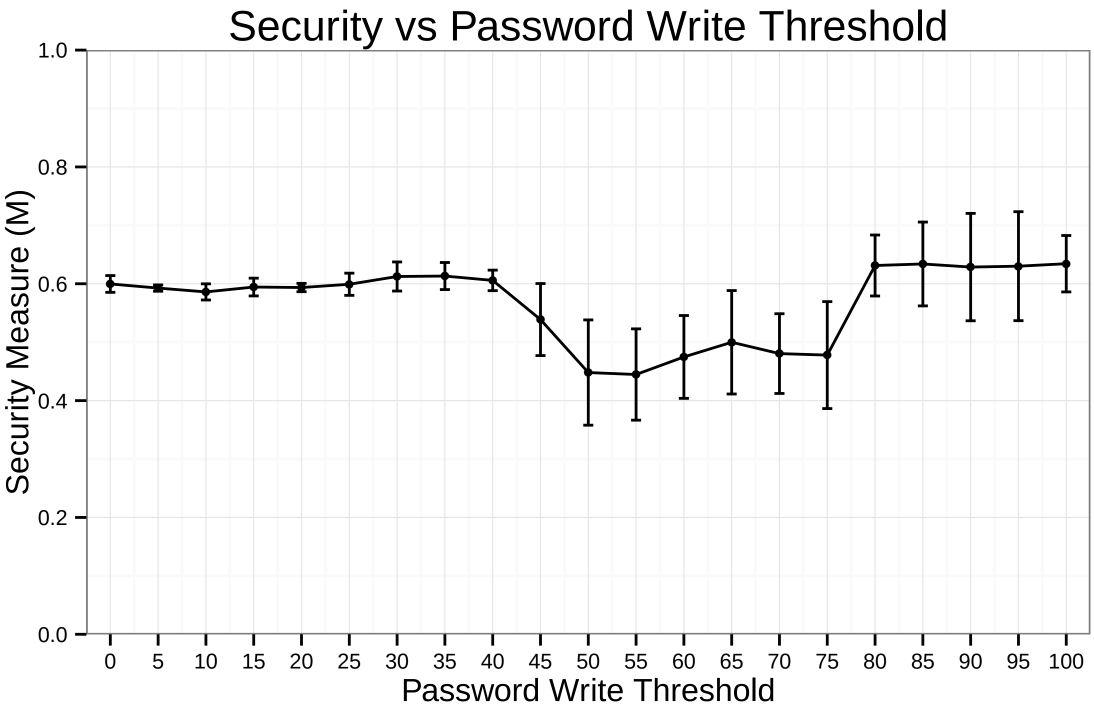


Figure 3.5: Security vs. password write threshold.

Distribution of Services. In Figure 3.9 we look at how changing the number of services while maintaining a fixed percentage of weak, average, good, and strong password composition rules for three pairs of cognitive threshold settings affects security. Each curve appears to reflect a sigmoid function flipped along the y-axis and shifted accordingly. This is what one might expect. For a small number of services users are able to simply remember their passwords without resorting to circumvention. As the number of services grow users circumvent.

In Figure 3.10 we use a fixed number, 24, of services and vary the distribution of password composition policies for the same three pairs of cognitive threshold settings. These cognitive threshold pairs correspond to the password reuse and password write thresholds respectively. For the low cognitive threshold pairs, (20/30) and (40/60), circumvention is rampant for most distributions; hence, simply having the most strin-

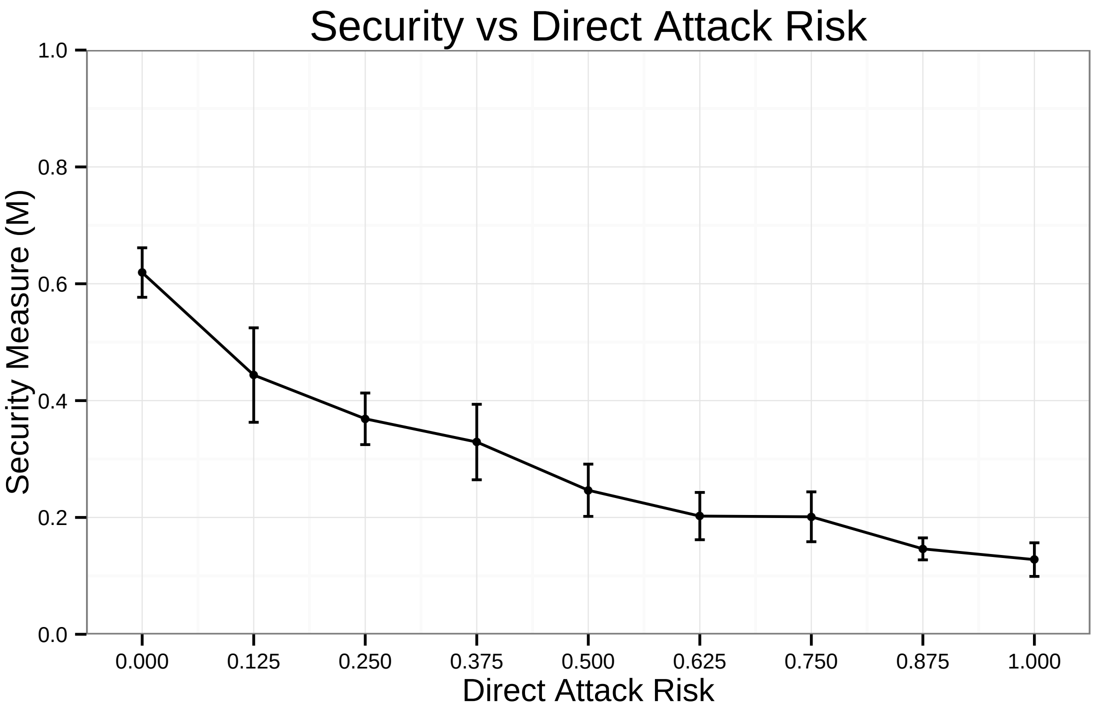


Figure 3.6: Security vs. direct attack risk.

gent password composition policies tends to make sense because the primary factor in our security measure becomes the raw password strength. For the highest cognitive threshold pair considered, (60/90), there's less circumvention; users may be able to choose a larger set of unique passwords for less stringent distributions, thereby reducing the likelihood of reuse attacks.

We believe further experimentation would demonstrate that even for low cognitive threshold pairs we achieve better security by using weaker distributions under different, but still viable, parameter settings (e.g, changing the password reuse attack risk from 0.25 to 0.5). We leave this for future work.

3.3.7. Takeaways

While we cannot make specific password policy recommendations based on our model, which requires further validation, we do believe our results provide some valuable

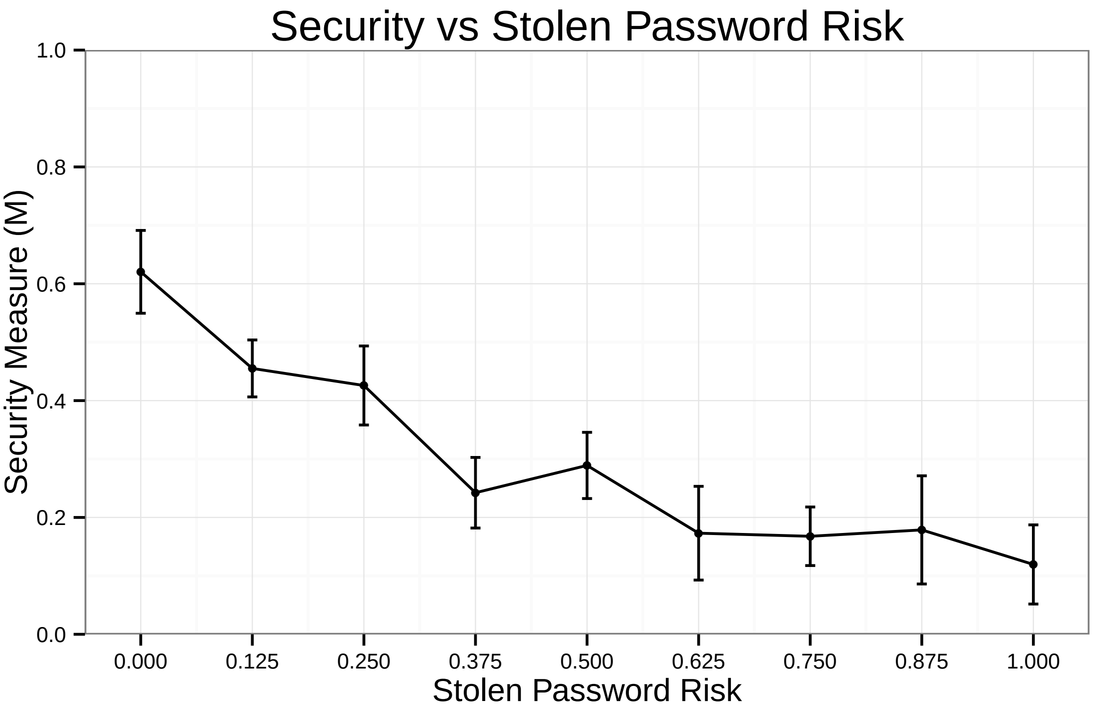


Figure 3.7: Security vs. stolen password risk.

insights that serve as indicators of how to improve password policies:

- Always choosing the most stringent password composition policy may be disastrous, endangering both usability *and* security with no gains.
- All circumvention is not the same. To improve security at a given organization, one must pinpoint the threat model and design policies accordingly.
- Endorsing relatively benign circumventions at an organization may reduce the prevalence of particularly malignant circumventions. As an example, it may very well make sense for an organization to give their employees a small card to write their passwords on if security personnel are more worried about a password reuse attack than an adversary stealing passwords from cards.

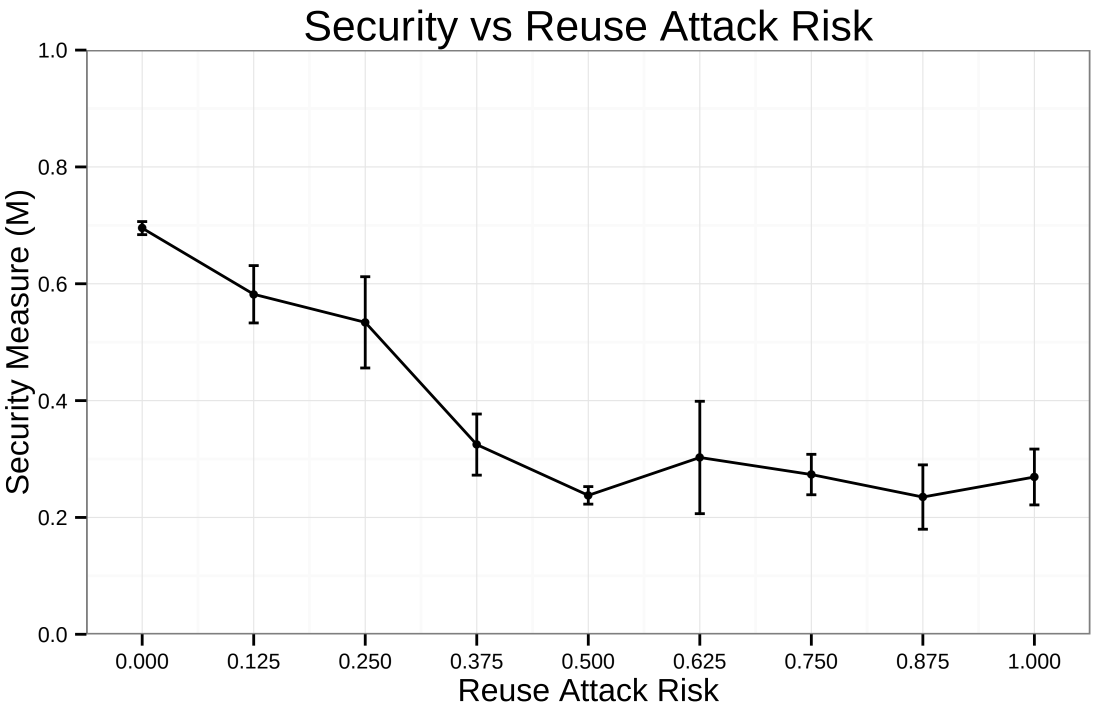


Figure 3.8: Security vs reuse attack risk.

Section 3.4

Future Work

While we feel there's a lot to be done in this space, primary foci for future work include adding to the password management model and building an agent-based model for an autologout scenario.

3.4.1. Password Management Scenario

We are interested in incorporating more faithful and/or better reasoned models and processes (e.g., [63])) for password recall, cognitive burden, and forgetfulness into our simulation. Once we've done this, we'd also like to revisit the work mentioned in this paper and explore other password management challenges. For a few examples, we'd like to (a) develop a more elaborate password simulation that incorporates commu-

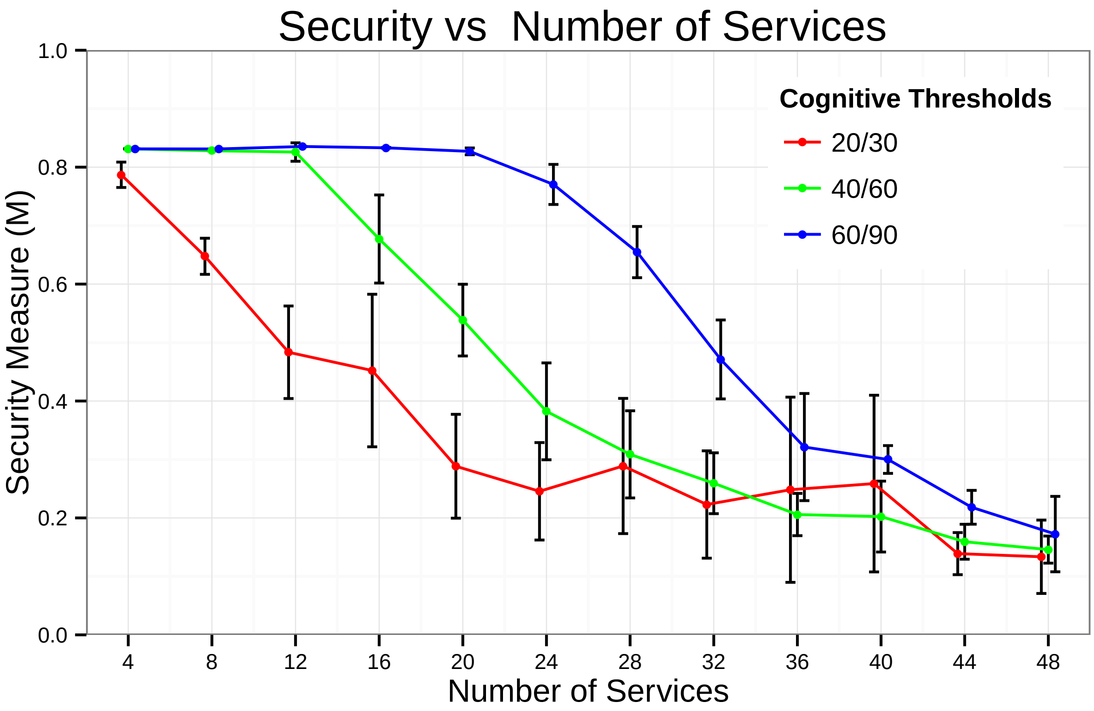


Figure 3.9: Security vs. number of services.

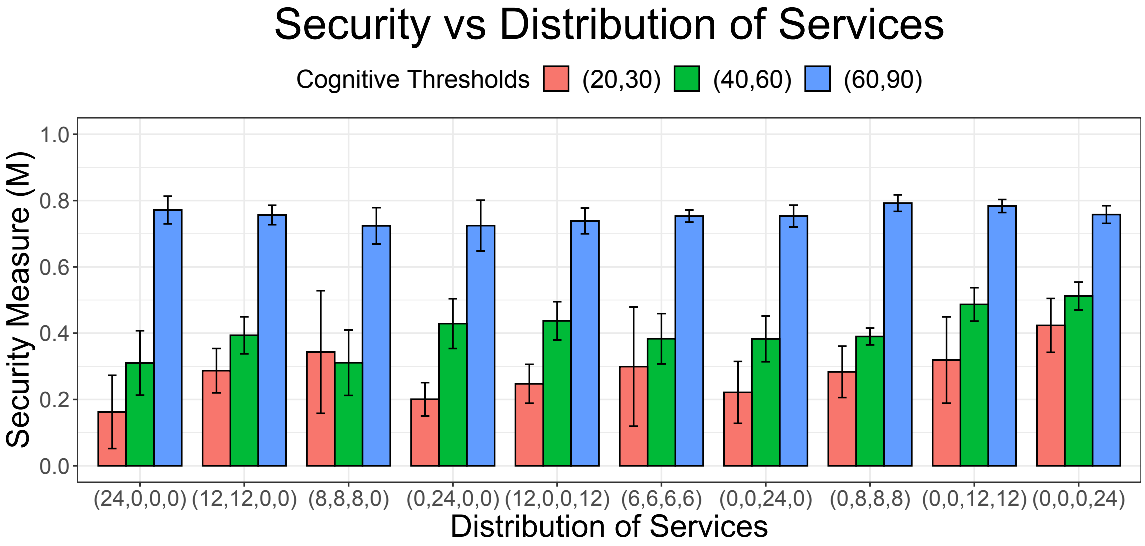


Figure 3.10: Security vs. distribution of services.

nication and password sharing between users, exploring how group dynamics affect circumvention, (b) model how users cope with enterprise requirements requiring them to frequently reset their passwords, or (c) test alternative password policies (e.g., what would happen if we allowed users to write passwords on Post-It notes for a limited duration of time, but told them to rip up the Post-It notes afterward?). The idea of recognizing and even incorporating existing circumvention into the security model also seems like an interesting pursuit for modeling work. Last, while we have tried to validate our work with previous studies, this is an ongoing challenge and we would like to pursue new avenues and perhaps devise new experiments to aid on this front.

We also note that two former Dartmouth undergraduates have spearheaded work, in collaboration with us, that builds on the password management simulation presented in this chapter. Bruno Korbar [99] worked on validating the simulation, and Christopher Novak et al. [128] used simulation to examine password memorization techniques.

3.4.2. The Effect of Security Policies on Group Behavior and The Autologout Scenario

Tackling even an ostensibly simple problem, such as setting a “good” timeout threshold, can be a nightmare in practice. On paper, the general shape of a timeout vs. security curve seems obvious: surely, it’s a monotonically decreasing curve! In practice, humans act according to flawed belief systems, they interact with other humans and other systems, they work toward achieving many competing goals, and they are plagued by deficiencies that lead to suboptimal decision-making, behaviors, and other phenomena; thus, we find the resulting curve can often be counterintuitive. Indeed, in a compiled corpus of circumvention scenarios we collected, we observed many examples of such *uncanny descents* where dialing up a security knob worsens aggregate security. [180].

Regarding the challenges of the timeout problem, consider the following anecdote. In a large hospital, clinicians frequently left shared computers logged-in but unattended [100]. Security officers, concerned about inappropriate access and inadvertent modification of patient data, opted to attach proximity sensors to the machines in an effort to mitigate these risks. These sensors detected when users had left terminals logged-in but unattended for some fixed timeout threshold. When such an event was detected, the logged-in user was automatically logged out of the machine. Clinician reception of these proximity sensors startled security officers. Clinicians, annoyed with the system, which was an impediment to doing actual work, placed styrofoam cups over the proximity sensors, which effectively tricked the proximity sensors into believing clinicians were nearby when they were not. The proximity sensors were an absolute failure. Resources had been spent with the goal of improving security, but doing so yielded no security gains; instead, it was utterly defeated and it probably created a greater rift between clinicians and security personnel, making future security challenges even more difficult to address.

This anecdote highlights that it is essential to find solutions that make sense in the context of enterprise workflow—solutions that can be successfully adopted by users while also realizing security objectives without sabotaging other objectives.

So, how do we arrive at these solutions? It is usually impractical for security personnel to test out different security approaches within existing enterprises. Even if it is feasible, doing so often involves, at the minimum, substantial time, implementation costs, maintenance costs, and depletion of a finite user compliance budget [20]. We contend that multi-agent simulations may help distinguish good solutions from bad ones by predicting stress points of candidate implementations, thereby suggesting things to improve upon. However, we are not suggesting that agent-based modeling is some magical panacea that can be used to address all security problems. It has its limits; it is nigh impossible to predict the unprecedented. Instead of trying to predict inventive workarounds such as the placement of styrofoam cups over proximity sensors, we believe simulation is an effective tool to gauge user inclination to circumvent.

We can estimate the risk from user circumvention in terms of motive, opportunity, and potential harm. Consider the case of auto-logouts. First, motive stems from the frequency of workers leaving and returning to shared workstations, where the time taken to log in becomes a significant drain when summed over many instances. Second, opportunity also arises from the shared environment, where workers might remain logged in to avoid these costs, or use another's credentials, inadvertently or not. Third, the potential harm comes from the nature of the task, since medication prescriptions or notes of delivery may then be ascribed to the wrong patient.

Using simulation, we can explore the relevant factors that affect security risks associated with a clinician using a terminal to which another clinician is logged in. The likelihood of risk is affected by the number of agents, the number of workstations, group attitudes towards security and circumvention, and the dynamic nature of tasks; the actual risk is affected by the kinds of tasks performed. Simulations allow us to compare how burdensome different kinds of solutions are on users. For example, we might compare an auto-logout solution to a solution involving authentication challenges after a period of inactivity, which may slightly reduce the burden of having to log back in to a service; or, we could detect tasks that are disparate from the current task and warn the user that they may be using a terminal to which someone else is logged in. For some tasks, it may be possible to predict whether the worker must return to complete their session, and to apply different policies based on this prediction.

Last, while we mentioned the timeout problem in the medical setting, there are numerous other scenarios where auto-logouts may be relevant. And, we believe modeling approaches could be developed for them as well.

Section 3.5

Conclusion

We have discussed our work toward building an agent-based model for a password management scenario. While validation is a challenge, we have made first steps toward building a useful cognitive behavioral agent-based model for password circumventions; we've also performed trials that have generated what we believe to be interesting and perhaps even counterintuitive results. For example, under certain assumptions, making password composition requirements more stringent may actually lead to a decrease in aggregate security. For another example, allowing users to write down passwords may actually improve security by reducing the likelihood of password reuse and reuse attacks. Password management is just one of many areas where we believe cognitive behavioral agent-based models can serve as a useful tool. In Smith et al [180], we observed that a pattern of policy choices at one site counterintuitively affects security at other sites. Applying agent-based modeling to these sorts of scenarios and others, such as those mentioned in Section 3.4 may provide useful insights that are otherwise difficult to attain.

Chapter 4

Human-Computability Boundaries

Human understanding of protocols is central to protocol security. The security of a protocol rests on its designers, its implementors, and, in some cases, its users correctly conceptualizing how the protocol should work, how it actually works, and how others will perceive how it works. Ensuring these conceptualizations are correct is difficult. A complementary field, however, provides some inspiration on how to proceed: the field of language-theoretic security (LangSec) promotes the adoption of a secure design-and-development methodology that emphasizes the existence of certain computability boundaries that must never be crossed during parser and protocol construction to ensure correctness of design and implementation. In this chapter, we discuss the idea of supplementing this work, which is grounded in classical computability boundaries, by taking into account human-computability boundaries. Classic computability research has focused on understanding what problems can be solved by machines or *idealized* human computers—that is, computational models that behave like humans carrying out rote computational tasks in principle but that are not subject to the natural limitations that humans face in practice. Indeed, as Kahneman and others have show in various domains, such as economics, psychology, sociology, and usable security, people do not always behave as we might expect. Humans are often subject to a variety of deficiencies, e.g., constrained working memories, short attention spans, misperceptions, and cognitive biases. We argue that such realities must be taken into consideration if we are serious about securing protocols. A corollary is that while the traditional computational models and hierarchies built using them (e.g., the Chomsky hierarchy) are critical in securing protocols and parsers, they alone are *inadequate* as they neglect the human-computability boundaries that define what humans can do in practice. In this chapter, we advocate for the discovery of human-computability boundaries, present challenges with precisely and accurately specifying these boundaries, and outline future paths of inquiry.

Section 4.1
Introduction

Humans are integral to the conception and operation of protocols. They lay out the initial vision, create the specification, implement the protocol, and wittingly or unwittingly make use of it. Due to humans' close and varied interactions with protocols during their design, development, and operation, we must—if we want to secure protocols—account for humans' intrinsic limitations in understanding protocols.[1]

The genesis of a protocol vulnerability often lies in some human failure or deficiency, e.g., the copy-and-paste blunder that produced the Apple *goto fail* vulnerability [124]. The designer may introduce mistakes or create the specification under incorrect assumptions. Or the implementor may fail to correctly conceptualize the specification, e.g., due to cognitive constraints. Or perhaps the user may misunderstand the protocol, driving them toward behaviors that jeopardize security. (While some may not consider the previous example to be a protocol vulnerability, it has the same form as one; it is a predictable failure of the protocol design-and-development

[1] While the discussion in his chapter focuses on protocols, the notion of human-computability boundaries is certainly applicable more broadly.

process, which can be used as a reliable conduit for attack.)

Our thesis is that a whole class of vulnerabilities could be averted if we better understood human limits to computability and took a principled approach to protocol design and development grounded in such an understanding.

In the remaining sections of this chapter, we: discuss Turing's notion of computability; provide a brief primer on the field of language-theoretic security (LangSec), which informs our work; present the idea of complementing LangSec with the incorporation of human-computability boundaries; discuss challenges in defining human-computability boundaries and follow-on work; discuss related work; and conclude.

Section 4.2

The Human Computer

Today, Turing machines are often thought of as computational models for modern-day electronic computers; however, Turing very much had humans in mind during his conception of the Turing machine. As Jack B. Copeland points out in his discussion on the Church-Turing thesis:

> "Turing introduced his machines with the intention of providing an idealized description of a certain human activity, the tedious one of *numerical computation*. Until the advent of automatic computing machines, this was the occupation of many thousands of people in business, government, and research establishments. These human rote-workers were in fact called *computers*. Human computers used effective methods to carry out some aspects of the work nowadays done by electronic computers. The Church-Turing thesis is about computation *as this term was used in 1936*, viz. human computation[.]" [47]

In Turing's seminal paper [188], in which he proved the *Entscheidungsproblem* is not, in general, solvable, he also introduced the Turing machine, along with the notion of computability. Turing wrote, in the paper, that: "Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book." In the same paper, Turing uses "the fact that the human memory is necessarily limited" as justification for the finite state property of Turing machines. [2]

Despite Turing's inspirations to model human computation, Turing machines are not adequate in fully capturing all aspects of human computation in protocol and program design, development, and use. It was never meant to do this. The Turing machine was a computational model that dealt with an ideal—a human in principle, not in practice. More importantly, human computation at the time was envisioned narrowly as rote processes carried out by humans. It was never intended to capture how humans design, develop, conceptualize, and use computer programs and protocols, in the fashion they do today. While we still have human computation in the present day, the role of humans and the tasks they perform are fundamentally different—and any computational models we use to capture human computation must reflect this reality.

Section 4.3
LangSec and Computational Models

Language-theoretic security (LangSec) [2] incorporates the theoretical insights offered by language theory, automata theory, and computability theory into a design-and-development methodology that averts common pitfalls responsible for producing numerous protocol and parser vulnerabilities. It advocates separating the parser from

[2] We note that not everyone held this view. For example, Shagrir provides discussion on Gödel's rejection of this assumption [170].

the execution environment, modeling the intended behavior of the parser as a formal grammar, ensuring the grammar does not exceed certain computability boundaries on an extended version of the Chomsky hierarchy, and ensuring that the parser is a *recognizer* or more precisely a *decider*, i.e., it rejects all bad inputs and accepts all good inputs. In essence, LangSec tells us how to design protocols and parsers based on our understanding of the limitations of machines. That is not to say that LangSec does not acknowledge or address human causes of protocol and parser vulnerabilities. On the contrary, Bratus et al. in their discussion of exploit programming [38], note that many exploits are manifestations of incorrect computability assumptions. LangSec aims to rectify these assumptions within the design-and-development process. Furthermore, successful application of LangSec principles *requires* reducing human error. For example, the parser combinator toolkit Hammer [141] helps eliminate user error by assisting the implementor in creating a parser that matches the specification grammar. We contend that, while LangSec is vital and has made great strides toward securing protocols, it alone is insufficient. Specifically, there is a limit to what can be achieved by considering traditional computability boundaries alone. (Of course, one might argue this would not be a problem if we could eliminate the human from all parts of the protocol life cycle, including design, development, and use; as far as we can tell, we're not quite there yet.)

We propose supplementing the field of LangSec with work that explores human-computability boundaries. Classical computational models, such as the Turing machine are excellent for capturing what machines can do; however, they are generally not well-suited for capturing what actual humans can do with and especially without aids. In practice, humans have finite memories and often inadequate knowledge to understand protocol workings in comparison to machines. They have short attention spans. They are subject to cognitive biases and often make mistakes in reasoning in predictable ways. These deficiencies manifest in bugs during protocol and parser conceptualization, coding bugs, and user error, all of which endanger security. [179]

We argue that we must acknowledge these human deficiencies, understand why and how they occur, develop solutions to begin addressing them, and finally we must update our protocol and parser design-and-development processes in accordance with such findings.

Section 4.4

Human-Computability Boundaries

Using an extended version of the Chomsky Hierarchy that differentiates between non-deterministic and deterministic pushdown automata, LangSec recommends staying within either the boundary of Turing-decidability (linear-bounded automata) or the stricter boundary of parser-equivalence decidability (deterministic pushdown automata), depending on the problem at hand. The *exact* class boundaries for these decision problems are not part of the five-class extended Chomsky hierarchy, e.g., the Turing-decidability boundary lies at recursive languages. The extended Chomsky hierarchy, however, is natural for humans to interpret and allows sufficient expressiveness to still be useful in the design and development of parsers and protocols.

Human-computability boundaries—the boundaries that specify what *actual humans* can do with the capabilities they possess and the deficiencies they are subject to—are a different beast altogether. Fitting human-computability boundaries to an extended Chomsky hierarchy is futile as there exist grammars within the class of regular grammars—i.e., grammars that can be expressed with finite state automata—that humans, in general, fail to conceptualize correctly. We do not know exactly where these human-computability boundaries lie, but the discovery of them may be instrumental in securing protocols and parsers. This observation is captured in Figure 4.1.

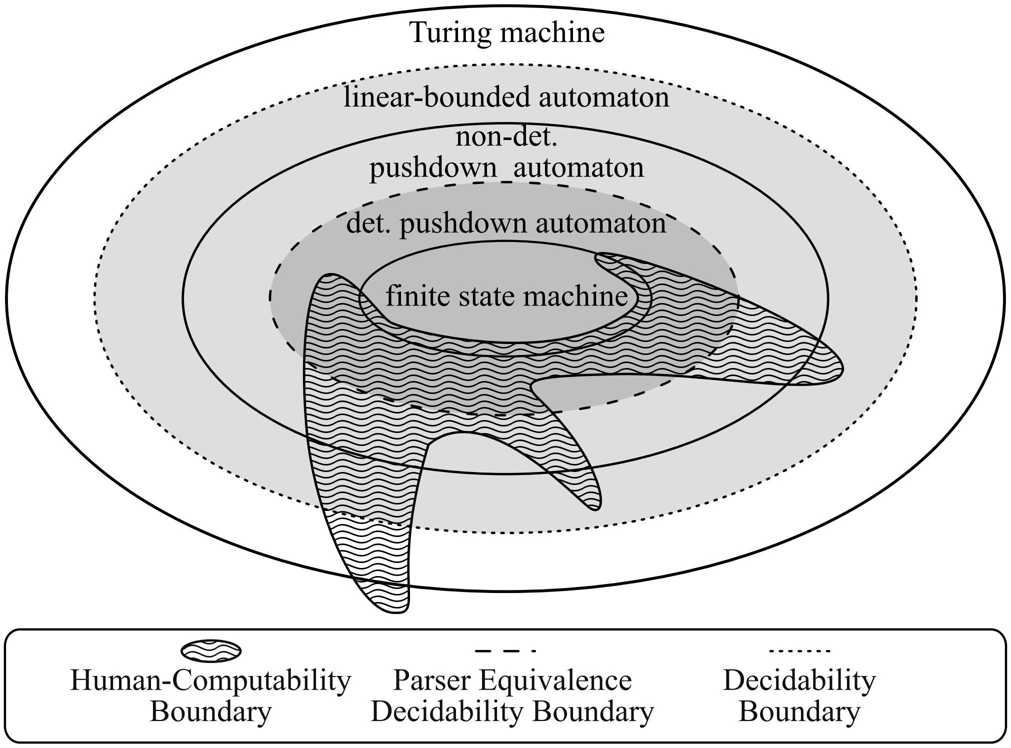


Figure 4.1: Human-computatability and LangSec boundaries.

The ovals correspond to classes of grammars (or languages or automata) in the five-class extended Chomsky hierarchy. LangSec boundaries are drawn at linear-bounded automata and deterministic pushdown automata, whereas the oddly-shaped blob corresponds to a single idealized human-computability boundary. ***If this boundary were representative of reality, we would want to constrain ourselves to the intersection of the blob and the appropriate LangSec computability boundaries during protocol and parser construction.***

In practice, however, things are more complex. We can imagine different human-computability boundaries corresponding to different human roles and protocol interactions. We can also imagine fuzzy boundaries where the uncertainty comes from the variance of human attributes over a subpopulation. We might consider human deficiencies of a probabilistic nature and aim to ensure most users are unsusceptible to a given flavor of attack based on protocol misconceptions; then, we may design and develop the protocol around this aim. If we know *a priori* what tools the various actors have at their disposal, the model we choose and boundaries we choose should take this into account. In short, the model used to express human-computability boundaries should be rooted in the protocol at hand, as well as the relevant subpopulations and their capabilities.

Section 4.5

Challenges and Future Work

In the previous section, we introduced the notion of human-computability boundaries and motivated the need for their discovery. However, there are a wide variety of challenges associated with accurately and precisely defining where these boundaries lie, developing models to capture them, and utilizing them in practice. In this section, we briefly touch on these threads and suggest directions for future research.

4.5.1. Determinants of Human-Computability Boundaries

There are many factors that determine where human-computability boundaries lie, e.g., memory, attention span, the dual-process model of cognition, and bounded rationality. [78, 179] However, some of these determinants will have a larger impact than others and some information will be easier to attain and utilize in addressing vulnerabilities that arise from human deficiencies. That is, pragmatically speaking, the utility of exploring a determinant rests on its *salience* with respect to human-computability boundaries and whether the information we can acquire about the determinant is *actionable*. The effectiveness of the models that enable us to determine where human-computability boundaries follows directly from the determinants we choose.

4.5.2. Usability Studies

Identifying the determinants of human-computability boundaries is insufficient. We must also conduct usability studies to understand the interplay between these determinants, human-computability boundaries, and security. Of course, this is not a one-way process; usability studies also help with identifying new determinants, which in turn guide new usability studies.

One example of a genre of usability studies we are interested in involves collecting concrete metrics for code complexity. Two classes of metrics are based on: (a) what the programmer can readily observe in the code and (b) what is represented in the abstract syntax tree (AST) for the program inputs in computer memory. As we mentioned earlier, program inputs are handled by code called parsers. Examples of metrics of the first type include lines of parser code and complexity per line of parser code, e.g., how many atomic structures such as combinators are used or represented in each line of code (on average or on the worst line). Examples of metrics of the second type include AST depth, number of branches, and tree balance.

4.5.3. Understanding Roles

Drawing useful human-computability boundaries requires understanding which roles are pertinent, the goals associated with the roles, the tools afforded by each role, and the interplay between each role and the protocol. Such understanding must be reflective of the protocol at hand and the application domain. The protocol and application domain may warrant consideration of additional roles or subroles that we have not discussed.

4.5.4. Developing Models

We've discussed the importance of defining where and how the protocol is used, determining the roles of the various human actors, identifying the determinants of human-computability boundaries, and gathering the requisite data grounded in usability studies to draw human-computability boundaries. The next step is then to incorporate these findings into a model that captures human-computability boundaries in a way that enables us to reason about the security of the protocol. It may be infeasible to draw perfect or even close-to-perfect boundaries for human computability. Understanding *some* limitations, however, can go a long way in addressing vulnerabilities.

The power of the model used to capture human-computability boundaries lies in its utility in the design and development of safe protocols. Even if we cannot perfectly capture human-computability boundaries, all is not lost. Indeed, it may be better to capture a few limitations in a manner that enables us to design and develop safe protocols than many in a way that does not. As we discussed earlier, one inspiration for this work is to develop human-computability boundaries that complement LangSec boundaries. In pursuit of this objective, we may wish to develop models similar to those of the classical automata, such as Turing machines, to capture these boundaries. While even these models will not neatly fit within the extended Chomsky containment hierarchy used in LangSec, they would still be rooted in automata theory, which is certainly convenient. After all, understanding the commonality of two models of one type is generally easier than understanding the commonality of two models of different types.

We note that there has been some interesting, recent work on developing models for end users (e.g., [29, 18, 88]) that can assist in safe protocol and program construction. Another approach might be to extend the compliance budget work of Beautement et al. [20] to a cognitive budget for human agents.

Section 4.6

Related Work

Jeanette M. Wing expounded on computational thinking as an essential mindset that everyone would benefit from, thereby providing a strong pedagogical basis for incorporating computational thinking into college and pre-college curricula [213]. She writes:

> “Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine’s instruction set, its resource constraints, and its operating environment.” [213]

This mindset is crucial in efficiently solving problems on machines. We argue for a parallel notion: Just as we must understand the computational capabilities of the machines that humans use, we must understand computational capabilities of humans as they interact protocols and programs, e.g., as they conceptualize and reason about code during development.

For completeness, we note that in recent years, human computation has developed into a field in its own right, e.g., [150, 111, 191]. The work in this field, however, is largely tangential to our work here. Our interests are in developing an understanding of human-computability boundaries as they pertain to secure program and protocol design, development, and use. That said, in the past two decades, there has been some exciting research efforts to capture humans in protocol and parser design. Below, we touch on a few particularly relevant ones.

In 2007, Carl Ellison[58] presented the notion of ceremony [3] as a natural extension to the network protocol. A ceremony incorporates everything conventionally thought to be out-of-band to the protocol, e.g., UI interactions, human-human interactions, provisioning tasks. This holistic view of the protocol as a ceremony enables the security practitioner to better conceptualize and analyze protocol security. Since then, researchers have expanded on the idea of ceremonies. Notably, Bella and Coles-Kemp [21] pursued a formal model of security ceremonies with multiple layers: information, operating system, human-computer interaction, personal, and communal.

Johansen et al. [88] argued for the development of a new discipline, Behavioral Computer Science, lying at the intersection of behavioral sciences, ubiquitous computing and Internet of Things (IoT), and artificial intelligence. This discipline blends the study of HCI, modeling, and the notion of computational trust. The authors argue we must rethink the rational agent models often used for human behavior by acknowledging that: differences exist between humans' experienced utility, predicted utility, and remembered utility [92]; humans employ the dual-process model of cognition wherein they may invoke either a fast, knee-jerk, intuitive, and automated response or a slower, deliberate, rational response [177, 91]; and humans are subject to all sorts of heuristics that affect their judgements [66]. The authors then discuss approaches to building models that capture this complexity, grounded in the

[3] As noted by Carl Ellison: "The term 'ceremony' was coined for this purpose by Jesse Walker of Intel Corporation." [58]

Bella-Coles-Kemp model discussed earlier [21].

Basin et al. [18] studied the security of protocols in the presence of human error. They developed a formal model that includes human agents whose behavior may deviate from the behavior assumed by the protocol specification. They captured human error using two approaches: (1) a skilled human approach that begins with an infallible human agent who knows the protocol specification and modifies it to allow for a small number of mistakes; (2) a rule-based approach that begins with an untrained human that does not know the protocol specification and imposes a set of rules upon human agent behavior that dictate permissible behaviors. They then demonstrate how these two approaches can be used to formally model fallible humans with the Tamarin verification tool [117]. They also do a case study to show how this modeling approach can be used to discover human-based vulnerabilities in a protocol, and they use the model to compare different authentication protocols.

The most relevant work we've seen is by Blum and Vempala [29]. They proposed a model of human computation for end users in studying the security of protocols. They argued that traditional notions of computability cannot blindly be applied to humans and that, instead, human computational models must take into account the reality that human processing power is inferior to that of computers. They argued that human computation occurs in two distinct phases: a pre-processing phase and a processing phase. Accordingly, they developed a model for human computation—a variant of the Turing machine—and introduced the notion of a schema to be the human analog to a computer algorithm. Finally, they applied this model to different problems. While there is certainly some overlap with our work, we explore notions of human-computability boundaries more generally. Additionally, we are not solely concerned with users; we also focus on human designers and implementors. Last, we are interested in combining models of human computability with traditional computability models.

Section 4.7

Conclusion

We argued that security rests, in large part, on acknowledging and accounting for human deficiencies in the design and development of network protocols. Existing LangSec work highlights theoretical computability boundaries along the extended Chomsky hierarchy for which the decidability and parser equivalence decidability problems are solvable. Staying within these theoretical computability boundaries is important for secure protocol and parser construction. However, they alone are insufficient. To realize the security properties designers and developers desire we must also consider the human-computability boundaries that define what humans can do in practice. In this chapter, we introduced the notion of human-computability boundaries, highlighted the difficulty in identifying them, and discussed open challenges for future work.

Chapter 5

Eyes on URLs: Relating Visual Behavior to Safety Decisions

Individual and organizational computer security rests on how people interpret and use the security information they are presented. A mismatch between the information that is trying to be conveyed and what the user interprets may result in user exposure to risks that endanger their own security and privacy, that of their organization, or that of the people serviced by their organization. In this chapter and the next one, we focus on one particularly challenging problem for users that has plagued both security practitioners and researchers for over two decades—that of determining whether or not a given URL is safe. In this chapter, we explore users' visual behaviors as they read URLs to gauge whether they are safe to click on. Eye tracking is not only a window through which we can understand users' visual processes; it also provides a glimpse into the underlying cognitive processes driving those visual processes. We report on a user study where 20 participants were tasked with classifying URLs as safe or unsafe while wearing an eye tracker that recorded eye gaze (where they look) and pupil dilation (a proxy for cognitive effort). Among other things, our findings suggest that: users have a cap on the amount of cognitive resources they are willing to expend on vetting a URL; they tend to believe that the presence of *www* in the domain name indicates that the URL is safe; and they do not carefully parse the URL beyond what they *perceive* as the domain name. Our findings can be used to improve security awareness training, to guide the construction of URLs that are easier for users to interpret, and to develop better defenses.

Section 5.1 Introduction

As people surf the web, check their email, and do other computer-related tasks, they interact with web addresses or Uniform Resource Locators (URLs) [199]. Unfortunately, URLs do not only serve legitimate content; bad actors may use URLs under their control to conduct attacks, e.g., to serve malware or steal credentials by masquerading as a legitimate service. Thus, users must be vigilant. Trusting an unsafe URL could present a security threat to the individual or their organization. Yet users don't want to ignore safe URLs either. This problem is compounded by user misperceptions of URL syntax, the sheer time required to vet URLs, and some practices of legitimate services (e.g., use of URL redirectors). These factors make it very difficult for users to vet URLs. Consequently, many attacks rely on the victim unwittingly clicking on a malicious URL.

From a security standpoint, it is critical to safeguard users from malicious websites. And so, numerous solutions have been developed. Some companies specialize in security training for users (e.g., [94, 147]). Others focus on limiting user exposure to unsafe URLs: Products and services like Microsoft Office 365 APT Safelinks [118] and Proofpoint URLDefense [148] check for malicious content served by URLs before allowing users to visit them. Some browsers similarly warn the user when they detect unsafe URLs (e.g., [122]). There is also abundant research on why users fall for URL-based phishing attacks (e.g., [52, 79]), on training techniques (e.g., [106, 120, 196]), and on defenses (e.g., [61, 114]), as well as other foci. However, to the best of our knowledge, this is the first study that solely focuses on understanding users' visual attention as they process URLs. Studying users' visual attention while processing URLs allows us to determine why certain attacks succeed, to measure the influence of URL characteristics on visual processing and cognition, and to determine the efficacy of countermeasures.

The work presented here serves as a first step toward developing a descriptive model of the relationship between URL characteristics and user visual behavior. We conducted a user study where users were asked to classify URLs as safe or unsafe while wearing an eye tracker. One key finding is that participants spent more time on processing URLs as URL length increased but only up to a point. Another is that participants relied more upon the `authority` component of URLs than any other component.

Section 5.2

Related Work

5.2.1. Eye Tracking and Reading

Eye tracking is considered to be a window into users' cognitive states [159, 98]. It has been employed to assess cognitive load [139, 140, 145, 218], reading strategies [28, 157, 83, 84], and design implications [68, 24]. We study users' eyes as they process URLs.

Users assess the safety of a URL by reading. The amount of visual attention given while reading reflects moment-to-moment cognitive processing [154, 218]. Researchers have sought to examine the relationships between reading and eye movements by using measures like fixations, saccades, regressions, and backtracks [174, 27]. Fixations are pauses in eye movements during which new information is acquired. Research has shown that users fixate longer while reading when “the processing load is greater” [90].

Reading and scanning text differs with respect to fixations and word skipping [156]. When and where someone looks next while reading is influenced by the reader’s ongoing mental processing [156]. Six commonly used eye-tracking measures are: fixation count, fixation count on various areas of interest (AOIs), proportion of time spent on each AOI, average fixation duration, fixation rate (fixation count/second), and gaze duration mean on each AOI [108]. We used all these measures, as well as pupil dilation and backtrack fixation count.

5.2.2. Pupil Dilation and Cognitive Load

As users read and evaluate URLs, they use cognitive resources. A common measure of cognitive load is pupil dilation [146, 107, 139]. When users face challenging tasks, their pupils dilate on the order of 0.1 to 0.5 mm [142, 19]. This task-evoked pupillary response (TEPR) indicates the cognitive load of the task. However, pupil dilation is also influenced by other factors like the amount of light entering the pupil (pupillary light reflex) [138, 142] and one’s emotional state [216, 181, 36]. To reduce these effects, we conducted the experiment in a windowless light-controlled room.

5.2.3. Neutral Mood Induction

Mood can affect a person’s ability to comprehend text and their judgment [64, 34]. Mood induction is used to understand and reduce the effect of mood [119]. Watching a film or a story is one of the most effective mood induction techniques [197]. To reduce the effect of mood and improve replicability, we had participants watch a video chosen to induce a neutral mood.

5.2.4. URL Security and Phishing

The turn of the century saw phishing—the act of masquerading as a legitimate entity to gather sensitive user information [201]—emerge as a leading attack technique. This led to newfound recognition of the security risks posed by malformed and obfuscated URLs. After all, phishing often involves tricking unsuspecting victims into clicking on malicious URLs that *look* safe. Though phishing techniques had been discovered and used earlier (e.g., AOHell [160]), phishing began garnering significant attention by the early-to-mid 2000s, when the global costs of phishing attacks skyrocketed to the hundreds of millions of dollars [201]. In response, security practitioners and researchers alike sought to combat phishing. Yet, despite valuable efforts, phishing remains a major security challenge. Here, we review the relevant literature and explain where the work presented in this chapter fits in.

Much literature is centered around understanding the factors that determine what makes a phishing attack successful, such as user knowledge of phishing and security indicators, user behaviors, and user susceptibility to different flavors of attack. In their seminal work, Dhamija et al. [52] conducted a user study wherein participants classified various legitimate and phishing websites. They found that many users did not understand or notice browser indicators (with 23% of participants relying solely on website content to determine legitimacy) and that spoofing browser indicators was easy to do and effective in tricking users. Wu et al. [214] found that security indicators were not a strong defense as users rarely checked or understood them. Downs et al. [54] presented findings from a pilot survey examining why users fall for phishing emails. Based on their findings, they argued that more effort should be spent on teaching users how to interpret security cues in browsers and ensuring these cues are easy to understand. Sheng et al. [173] studied the effectiveness of phishing training materials and the demographic determinants of phishing email susceptibility.

They found women were more susceptible than men, the 18 to 25 year old age group was most susceptible, and educational training could reduce phishing susceptibility by 40% but with increased misclassification of legitimate emails. Hong et al. [79] also conducted an email classification study; they found that gender, trust, and some personality traits correlated with phishing susceptibility. Goel et al. [67] studied the impact of psychological manipulation on phishing susceptibility by sending out fake phishing emails to students and found that contextualizing emails to induce fear of loss or anticipation of gain was effective. Benenson et al. [23] studied the effect of communication medium on spear phishing susceptibility. They sent students fake phishing messages via email and Facebook with a URL purporting to contain pictures from a party; click-through rates were 42.5% for Facebook and 20% for email. An earlier experiment that was similar by the same group [22], however, found a click-through rate of 38% for Facebook and 56% for email; the authors hypothesized the difference was due to addressing recipients by first name in the earlier study. Researchers have also categorized different types of phishing and URL obfuscation techniques, e.g., [55, 134]. That said, there are (ostensibly) legitimate reasons to obfuscate URLs or otherwise break user expectations of where URLs will take them, e.g., as URL redirection [203] or unobtrusively tracking users by modifying URLs on click [48].

As phishing attacks rely on user deception, many seek to educate and train users. Stockhardt et al. [184] compare the efficacy instructor-based, computer-based, and text-based training. Kumaraguru et. al [105] present and evaluate an embedded training system: participants were sent fake phishing emails; participants who were phished were immediately presented with an intervention response as either text and graphics or a comic strip (the latter being more effective). Games have also been heralded as an engaging way to train users. Sheng et al. [172] presented the game

Anti-Phishing Phil to train users to recognize and avoid phishing emails, which outperformed more traditional training techniques. Arachchilage et al. [14] documented the design and development of a prototype of a mobile game to train users to better classify URLs; classification accuracy improved from 56% to 84% after playing the game. Wen et al. [196] recently developed a role-playing email classification game, modeled after the popular document-vetting video game *Papers, Please* [200], to assist users in vetting emails; they found it to be more engaging and effective in teaching users to classify emails than both Anti-Phishing Phil [172] and the anti-phishing email training materials Barracuda PhishLine [17]. Companies also offer security awareness training and phishing simulations as products or services [94, 144, 143, 147, 165, 164]. There even exists a "12-episode video series [with] a compelling story, an incredible cast, and very high production values" that "makes learning how to make smarter security decisions fun and engaging." [95].

Many defensive measures have also been pursued. Egelman et al. [57] conducted a lab study to compare active and passive phishing warnings to defend against spear phishing. Active warnings were found more effective with 79% of warnings being heeded. Maurer et al. [114] found that displaying in-context security information unobtrusively was both acceptable by users and effective. Some defenses have been incorporated into products. Email filtering is common. Applications and services, including browsers, protect warn users when they enter malicious or risky URLs, e.g., [122]. Microsoft Office 365 ATP SafeLinks [118] and Proofpoint URLDefense [148]) are two services that detect and vet URLs before serving them to users. However, such techniques are not foolproof. For one example, Nathaniel presents an open redirect vulnerability existing on Google that was also used to circumvent Office 365 Safelinks [125].

Some may argue there is limited utility in understanding how users parse and classify URLs due to recent techniques that reduce or obviate the need for user involvement in the URL-vetting process. For some examples, browsers, applications, and services blacklist malicious URLs, detect malicious or risky URL constructions, and detect and vet malicious content served via URLs (e.g., Microsoft Office 365 ATP SafeLinks [118], Proofpoint URLDefense [148]). However, such techniques are not foolproof; for one example, Nathaniel shows how an open redirect vulnerability existing on Google could be used to circumvent Office 365 Safelinks [125]. Additionally, though these safeguards exist, they are by no means universally adopted; many users must still regularly vet URLs. Moreover, aside from any direct practical utility this study has for the URL-vetting problem, its findings may be instructive in developing and refining solutions to problems that involve users interpreting and utilizing security information more broadly.

Recently, there has been growing interest in using eye trackers to examine and improve user security behavior. Miyamoto et al. [120] developed an eye-tracking based system that trains users to look at the status bar. Xiong et al. [215] studied the efficacy of domain highlighting and intructing users to look at the address bar by conducting two studies, one involving an eye tracker in which the address bar was treated as a single area of interest. Alsharnouby et al. [8] conducted a study wherein they asked participants to classify websites, not just URLs, as legitimate or illegitimate while wearing an eye tracker and examined how users gauge website legitimacy and the effectiveness of security indicators. While similar in spirit to these studies, we focus exclusively on understanding how users process URLs. That is, we are working at a different level of granularity and are not concerned with how people visually process websites as a whole but specifically how they visually process URLs. This finer level of granularity enables us to dissect URLs into different parts and examine how people process each part. We seek to understand what parts of a URL people pay attention to, what parts they don't, when people give up, and how their eyes process different flavors of URLs, amongst other things.

5.2.5. A Brief Introduction to URL Structure

A uniform resource locator (URL) is a string of characters that specifies the location of a web resource and how to access it [199]. The original URL specification details URL structure [25]. Here, we present the bare essentials of URL structure at an appropriate level of granularity to understand our work.[1]

Each URL in our corpus has the form:

<`scheme`>: // <`authority`><`rest`>

The `scheme` component [25, 26, 198] corresponds to the scheme name, which specifies how to interpret the text following the colon. Common schemes are *http*, *ftp*, and *file*. Every URL in our corpus uses the *https* scheme.

The `authority` component specifies a subset of the host, port, username, and password [26, 198]. For URLs in our corpus,, the `authority` component has either the form `host` or `user@host` where `host` represents the host and `user` represents the username. In this study, the host is always a fully qualified domain name (e.g., *`www. wikipedia. org`*), that is, "a sequence of domain labels separated by '.' " [25]. The last domain label is the top-level domain. For URLs in our corpus, the `authority` component comprises everything following the leading *https://* until either the next /, if present, or the end of the line.

We call the last component `rest`, a catch-all term that is *not* borrowed from any specification or standard. It captures everything following the `authority` component. The `rest` component includes the path [25, 26, 198], which may be empty; it may also

[1] A more thorough treatment of URLs can be found in URL and URI specifications and standards [25, 26, 198].

scheme	*delims.*	authority	rest
https	://	www.google.com	/forms/about/

Table 5.1: Disaggregation of a URL into its three components.

include queries, fragments, and accompanying delimiters [25, 26, 198]. For every URL in our corpus, if the `rest` component is non-empty, it includes a path that "[identifies] the resource within the scope of [the] scheme and authority" [26], it begins at the first / character following the `authority` component, and it is the last part of the URL. Table 5.1 provides an example of a URL disaggregation into these three components. Please note the formatting style used for these components. Later, we define areas of interest of the same names but different formatting styles.

Section 5.3

Study Outline

Our long-term goal is to understand users' visual behaviors (and the underlying cognitive processes they manifest) as they process, interpret, and operationalize security information (including information embedded in URLs) when making security decisions. Identifying which factors affect visual behavior and how they affect it is vital in informing security solutions. Such information can be used to improve security awareness training or to better design user interfaces that aid in decision-making.

The work presented in this chapter is one step towards this long-term goal. We aim to capture how some URL properties affect visual behaviors. We attempt to control for other factors, but we do not explore them in this initial study. We propose hypotheses pertaining to how various aspects of a URL affect visual processing of the URL, test these hypotheses, and observe trends in users' visual behaviors.

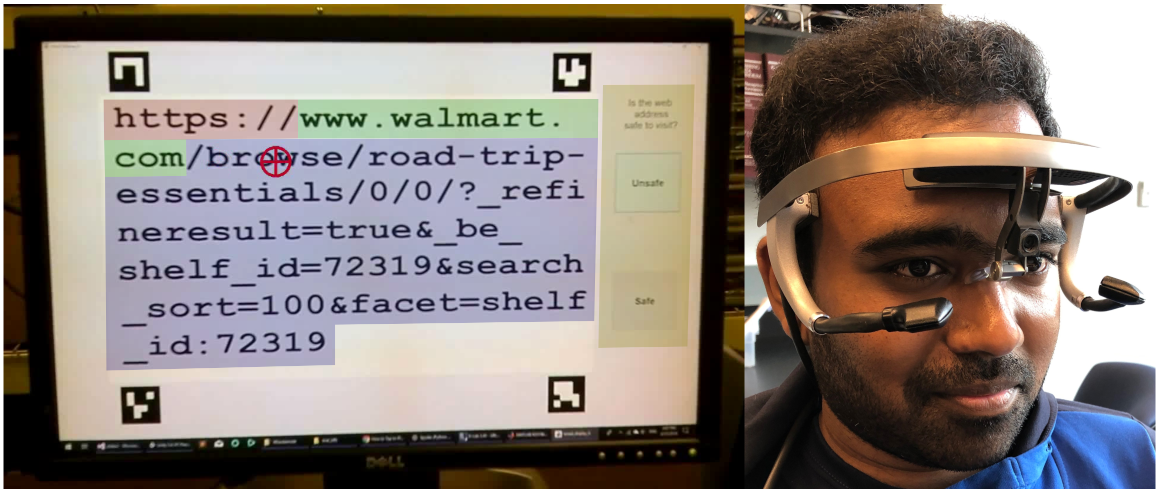


Figure 5.1: The left side of the figure is a processed frame from the eye tracker video (This is not the same as what the participant sees). The red cursor indicates gaze position and the four colored boxes represent four AOIs: the ***scheme AOI*** (red), the ***authority AOI*** (green), the ***rest AOI*** (blue), and the ***response AOI*** (yellow). The right side is an image of a participant performing the task wearing the eye tracker.

5.3.1. Hypotheses

We created hypotheses to examine how users visually process URLs and how URL features affect this processing:

H1: Total time spent on processing a URL is longer for complex URLs than it is for simple URLs.

H2: Total time spent on processing a URL, normalized by the URL length, is shorter for complex URLs than it is for simple URLs.

H3: There exists a URL length threshold over which increasing URL length does not result in more time being spent on processing URLs.

H4: Total time spent on the `scheme` per character is less than that of the `authority` and `rest` components.

H5: For URLs that have an `authority` component of form user@host where user ends with ".com", participants spend significantly more time per character looking at the user component than the host component.

Section 5.4

Method

5.4.1. URL Corpus and Classification

We created a URL corpus comprising 64 URLs partitioned into 8 categories.[2] Categories are defined by features corresponding to (1) safety, (2) complexity, (3) a leading *www* in the authority component, and (4) the attack type for unsafe URLs. The corpus contains 8 URLs for each of the 8 categories. To reduce variability and maintain uniformity between categories, every URL uses *https* as the scheme component and *com* as the top-level domain.

The categories are defined by the following 4 features:

Safety:. URLs that are *safe* use domain names associated with popular services within the USA, such as Facebook. We selected the fully qualified domain names used in these URLs primarily from the top 1,000 US websites in the Quantcast Top One Million list[3], although we consulted other lists as well. For the subset that were complex and included rest components, we chose the rest components by searching for legitimate content served by these domain names.

URLs that are *unsafe* have fully qualified domain names that, at the time of corpus construction, were eligible for purchase, did not have a domain name server record, or were spoofed websites. While many URLs with the *unsafe* feature were not actually unsafe to visit, it is exceedingly unlikely that participants would be knowledgeable

[2] Materials used in this study can be found at https://drive.google.com/drive/folders/1ZNMLoXBxOU4R2nela-6d7MxsaQGrdyg4

[3] https://www.quantcast.com/top-sites

about the status of the URLs tagged as *unsafe*, and, if an adversary wished to acquire the corresponding domains, they could do so. This decision allowed for greater control over the corpus.

Complexity:. URLs were grouped into two complexity classes: *simple* and *complex*. We define complexity in terms of (a) URL length and (b) URL features. A URL is *simple* if it is at most 36 characters long and does not contain a `rest` component. A URL is *complex* if it is at least 48 characters long and contains a non-empty path; it may also contain queries and fragments.

Presence of www:. URLs with the *www* attribute begin with ***`https://www`***. URLs with the *non-www* attribute do not.

Attack Type:. We chose to explore four conditions for unsafe URLs. They are neither exhaustive nor fully representative of real-world attacks. Rather, our aim was to explore a variety of conditions that may affect visual behaviors and/or classification:

- ***positive*:** The fully qualified domain name contains positive or feel-good words or phrases, e.g., “happy”, “bliss”.
- ***negative*:** The fully qualified domain name contains words or phrases with a negative, technical, or a security connotation, e.g., “malware”, “antivirus”, “techsupport”.
- ***substring*:** The fully qualified domain name has the form ***`https://X.Y.com`*** where ***`https://X.com`*** is a safe URL.
- ***user@host*:** The `authority` component has form *`www.X.com@Y`* where ***`https://www.X.com`*** is a legitimate URL. Moreover, some of the last four charac-

Category	Safety	Complexity	www	Attack Type
C1	*safe*	*simple*	*www*	N/A
C2	*safe*	*simple*	*non-www*	N/A
C3	*safe*	*complex*	*www*	N/A
C4	*safe*	*complex*	*non-www*	N/A
C5	*unsafe*	*simple*	*www*	*positive*
C6	*unsafe*	*simple*	*www*	*negative*
C7	*unsafe*	*complex*	*non-www*	*substring*
C8	*unsafe*	*complex*	*www*	*user@host*

Table 5.2: A summary of the 8 URL categories.

ters of Y are obfuscated using a hexadecimal representation, e.g., representing "*.com*" as "`.%63o%6D`".

The eight URL categories are presented in Table 5.2. In Section 5.4.5, we will discuss the measures in this table.

5.4.2. Experimental Design and Task

We conducted a within-subject experiment that was approved by the University of New Hampshire Institutional Review Board (IRB). Each of the 20 participants were shown the 64 URLs from the corpus over two sessions. The task was to classify each URL as safe or unsafe. Participants completed this task by viewing one URL at a time and clicking a button on the GUI to indicate whether they believed the URL was safe.

The URL corpus was split into two equal-sized sets presented over two sessions, such that four URLs from each category were represented in each set. For each session, the order in which URLs were presented was randomly determined but held fixed for all participants. However, session order alternated between participants.

5.4.3. Data Collection, Processing, & Analysis

We discuss the participant selection, the GUI, data collection, data processing, and data analysis:

Participants:. We collected data from 20 participants (3 female, mean age = 22.68, SD = 2.65). All participants were students who participated in the user study as part of their coursework. We discarded data from 4 participants due to technical issues with the data extraction from the eye tracker. Hence, we report on the data from 16 participants (2 female).

User interface:. The application was created using GUIs in MATLAB. It was presented to participants on a 24" monitor with a resolution of 1920x1200. Each URL image was created using bold monospace font [208] of size 64. The screen was made up of two panes. The first included the URL image, which was scaled and displayed on screen over 2-7 lines with a full line having approximate height of 20mm and width of 280mm. The second pane included the question "Is the web address safe to visit?", accompanied by two response buttons that read "Safe" and "Unsafe" (see Figure 5.1). Four markers were embedded in the application to identify the surface plane to mark various AOIs during post-processing of the eye-tracking data. Times of clicks and corresponding classifications/responses captured via button clicks were also recorded.

Eye Tracking:. We used the head-mounted Dikablis eye tracker to collect gaze positions. It contains three cameras: two eye cameras sampling the eye at 60 Hz and a scene camera sampling at 30 Hz. Gaze positions are computed from the pupil movements and mapped onto the video from the scene camera. Establishing a mathematical mapping between the features of eye and the target being looked at is referred to as calibration. We used the four-point operator-controlled calibration method [129].

Post-task questionnaire:. Following the URL classification task, the participant filled in a questionnaire comprising: demographics questions; questions pertaining to security knowledge and behaviors, especially regarding URLs and phishing; and questions to help assess experimental validity.

Data Analysis:. We used MATLAB for post-processing the eye-tracking data. We used JMP Pro 14 and R for statistical analysis. The Shapiro-Wilk test indicated that all of our data were non-normally distributed, thus we used non-parametric tests (Kruskal-Wallis test and Wilcoxon test) for analysis.

5.4.4. Procedure

After signing the consent form, the participant was given a brief introduction to the study and the user interface. They then saw a short neutral mood induction video to control for the effects of mood. They then filled in a pre-task questionnaire to assess their mood [167], wore the eye tracker, and completed a practice trial to familiarize themselves with the task and the GUI.

Before calibration, we adjusted a nose pin and head band to reduce the movement of the eye tracker during the study; we did not use a chin rest. Next, we focused the eye and scene cameras and calibrated the eye tracker using the four-point operator-controlled calibration method. The participant then classified URLs for the first session and took a break. The calibration procedure was then repeated and the participant classified URLs for the second session. Last, they filled in the post-task questionnaire. The distance between the screen and the participant was kept at about 0.6 meters.

scheme AOI	***authority AOI***	***rest AOI***
https://	www.google.com	/forms/about/

Table 5.3: Disaggregation of a URL in accordance with the first three AOIs. This differs from Table 1 in that the scheme AOI includes the "://" following the scheme.

5.4.5. Measures

Mood:. Each participant's mood was assessed along six emotional states: awake, pleasant, angry, fearful, happy, and sad [119]. The assessment used a 10-point scale, where 1 indicated that the participant's mood was not associated with the given emotional state, and 10 indicated that it was highly associated.

Score:. The score represents the number of correctly classified URLs within a set with no penalty for incorrect classification.

Total Time Spent:. The total time spent on classifying a URL is the time (seconds) from the presentation of the URL to the time when the user clicks on a button to classify it. This is a proxy for the cumulative effort and engagement in classifying the URL.

Time Spent on Areas of Interest:. Using the UTC timestamps of each data point recorded by the eye tracker, we computed the percentage dwell time on five AOIs (Areas of Interest). These measures express the distribution of users' visual attention and help us understand which URL components users use to gauge URL safety. We examined five AOIs. Figure 5.1 captures the first four AOIs and Table 5.3 gives a disaggregation of a URL in accordance with the AOIs that correspond to the URL. We now present the five AOIs.

- The ***scheme AOI*** captures the `scheme` component and the delimiters immediately following it. As every URL in our corpus uses the *https* as the `scheme`, this AOI always corresponds to the leading *https://* in the URL.

- The ***authority AOI*** captures the authority component. For classes C1 through C7, the authority component is a fully qualified domain name, e.g., *www.google.com* is the authority component of *https://www.google.com*. For class C8, the authority component has form user@host, e.g., as in *www.google.com@evil.com*. To test ***H5***, the ***authority AOI*** was further split into two smaller AOIs, the ***user AOI*** and the ***host AOI*** corresponding to the user and host components.

- The ***rest AOI*** captures the rest component.

- The ***response AOI*** captures the response portion of the screen containing the "Safe" and "Unsafe" buttons.

- The last AOI captured visual targets other than the previous four areas of interest.

Fixations and Backtracking Fixations:. Fixating is the act of maintaining one's gaze at a particular target for a certain duration of time. It represents the time where new information is gathered [152]. We extracted fixations of 100ms or more following prior research guidelines [163, 86, 123].

Backtracking is the process of revisiting information that was previously processed or skipped [40]. It usually occurs to re-establish previously processed information or it signifies a cognitive interest in an area with respect to the given task [42]. We measured the backtrack fixation count, i.e., the number of fixations involving backtracking.

Normalized Pupil Area:. : The eye tracker records raw pupil area of both eyes in pixels. We used the right eye pupil area. We used the Hampel identifier technique to remove outliers [Foroughi et al.2017; Pearson et al. 2016]. Due to the non-uniform sampling rate, we interpolated the data to obtain a uniform sampling frequency of 60 Hz [Pfleging et al. 2016]. Then, we normalized the data to compare it between participants.

Accounting for Length Differences in URLs:. URLs may differ in the number of characters in their `scheme`, `authority`, and `rest` components. Thus, for the corresponding AOIs, we calculated the time spent per character (total time spent on AOI divided by number of characters in AOI) and the fixation count per character (total number of fixations occurring on AOI divided by total number of characters in AOI). For the overall comparison, we computed overall time spent per character (total time spent/total URL length), overall fixation count per character (total fixation count/total URL length), and backtrack fixation count as a function of URL length (total backtrack fixations/total URL length).

Section 5.5

Results

5.5.1. Mood Induction Measures

On average participants were awake (ranking of M=7.50, SD=1.59), felt relatively pleasant (M=7.69, SD=1.40), and were mildly happy (M=6.75, SD=1.44). They did not feel angry (M=1.81, SD=0.83), fearful (M=1.56, SD=1.09), or sad (M=1.50, SD=0.82).

5.5.2. Scores

The average score was 40.44 out of 64. From the post-task questionnaire, we were able to identify whether the participants knew of the services associated with the *safe*

Probabilities	P[correct|known]	P[correct|unknown]
C1 (simple, www)	0.92	0.63
C2 (simple, non-www)	0.83	0.19
C3 (complex, www)	0.76	0.5
C4 (complex, non-www)	0.58	0.46

Table 5.4: Probabilities of correctly classifying *safe* URLs given the participant knew of the service.

Category	**URL Length**	**Time Spent**	**Score**	**Fix. Ct.**	**Back. Fix. Ct.**
C1	25.0 (4.8)	4.1 (2.3)	7.2 (1.1)	7.9 (4.9)	1.9 (1.8)
C2	19.8 (2.0)	4.0 (1.9)	3.8 (2.3)	7.1 (4.3)	1.6 (1.5)
C3	124.0 (13.2)	7.5 (3.8)	5.8 (1.5)	15.3 (8.2)	3.7 (3.1)
C4	105.3 (13.5)	7.9 (4.2)	4.4 (1.5)	15.9 (9.0)	4.1 (3.8)
C5	28.5 (2.4)	5.4 (2.1)	4.4 (2.8)	9.5 (4.7)	2.4 (1.9)
C6	29.3 (3.6)	4.8 (1.9)	5.9 (2.2)	9.2 (4.9)	2.4 (2.0)
C7	96.0 (20.4)	7.4 (4.0)	5.5 (2.1)	14.5 (8.3)	3.7 (3.2)
C8	95.0 (17.4)	6.3 (3.4)	3.4 (2.4)	12.6 (7.4)	3.2 (3.2)

Table 5.5: Mean values and standard deviations of measurements for the eight URL categories (not normalized by length). Measurements include URL length, time spent, score, fixation count, and backtracking fixation count.

URLs. Table 5.4 indicates the probabilities of participants correctly classifying the URL given that they knew the service. The Kruskal-Wallis test showed no significant difference between the four categories of *safe* URLs (C1-C4) in terms of the participant knowing the services associated with the domain names [$X^2(3)$=6.9674, p=0.0729].

5.5.3. Overview of Eye-Tracking Results

Table 5.5 presents some key results. The overall distribution of visual attention on the AOIs is shown in Figure 5.6. Using Kruskal-Wallis test, we found that the time spent per character was significantly different between the three AOIs corresponding to the URL [$X^2(2)$=30.4152, p<0.0001]. Post hoc analysis indicated time spent per character on the ***authority AOI*** was significantly higher than that of the ***scheme AOI*** and that of the ***rest AOI***. The fixation count per character was significantly different between the three AOIs [Kruskal-Wallis test: $X^2(2)$=23.9356, p<0.0001]. Post hoc analysis indicated that fixation count per character on the ***rest AOI*** was significantly lower than the other two. However, we found no evidence that fixation duration was significantly different between the three AOIs [Kruskal-Wallis test: $X^2(2)$=3.1692, p=0.0516].

The Kruskal-Wallis test indicated a significant difference in normalized pupil area [$X^2(2)$=8.7532, p=0.0126]. Post hoc analysis indicated a lower pupil area for the ***scheme AOI*** relative to other AOIs, suggesting less cognitive effort was expended on the ***scheme AOI***.

5.5.4. Complexity

We saw a significant difference in overall time spent (seconds) processing between *complex* and *simple* URLs [Wilcoxon test: Z=3.4865, p=0.0005]. More time was spent on *complex* URLs (M=7.26, SD=2.41) compared to *simple* URLs (M=4.58, SD=1.35). This can also be seen pictorially in Figure 5.4. Wilcoxon test indicated significant differences in overall time spent per character [Z=8.9998, p<0.0001], overall fixation count per character [Z=6.4883, p<0.0001], and backtrack fixation count as a function of URL length [Z=4.4399, p<0.0001].

People spent less time per character on *complex* URLs (M=0.06, SD=0.01) than *simple* URLs (M=0.13, SD=0.04). Figure 5.2 shows the time spent per character decreases as URL length increases. Also, the fixation count per character was smaller for *complex* URLs (M=0.12, SD=0.04) than for *simple* URLs (M=0.22, SD=0.10). Figure 5.3 shows a decrease in fixation count per character as URL length increases. But the backtrack fixation count was higher on *complex* URLs (M=3.68, SD=2.44) relative to *simple* ones (M=2.08, SD=1.18). We found no significant difference in the score between *complex* (M=4.76, SD=2.10) and *simple* URLs (M=5.34, SD=2.51). Examining *complex* URLs of different lengths tells a more nuanced story. Figure 5.5

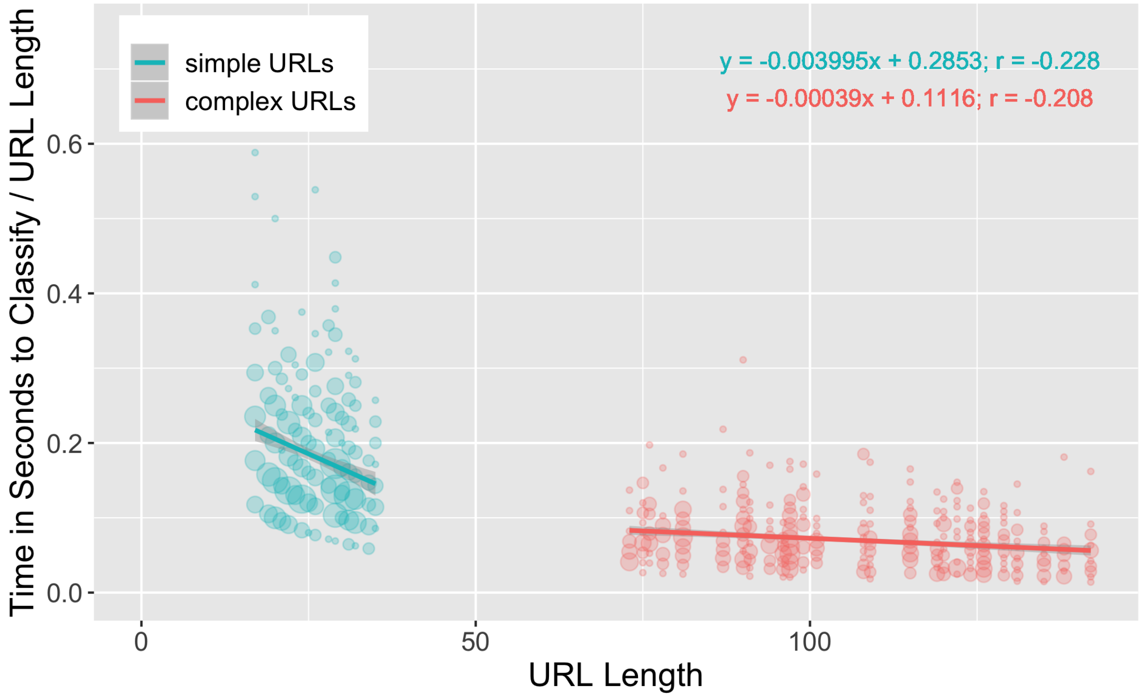


Figure 5.2: Time spent per character to classify URL vs. URL length with linear regression lines. As URL length increases, participants spent less time per character on classifying URLs overall. This suggests that the amount of effort people are willing to invest reduces as you increase URL length.

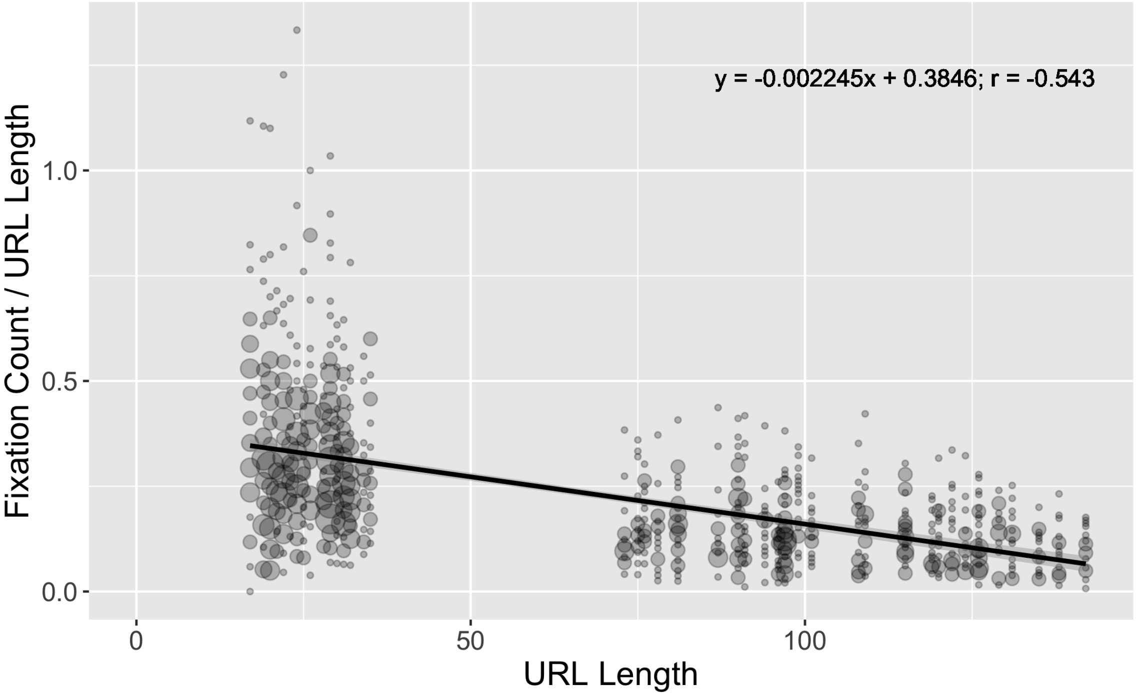


Figure 5.3: Fixation count per character vs. URL length with a linear regression line. As URL length increases, the fixation count decreases, which suggests a lower cognitive load.

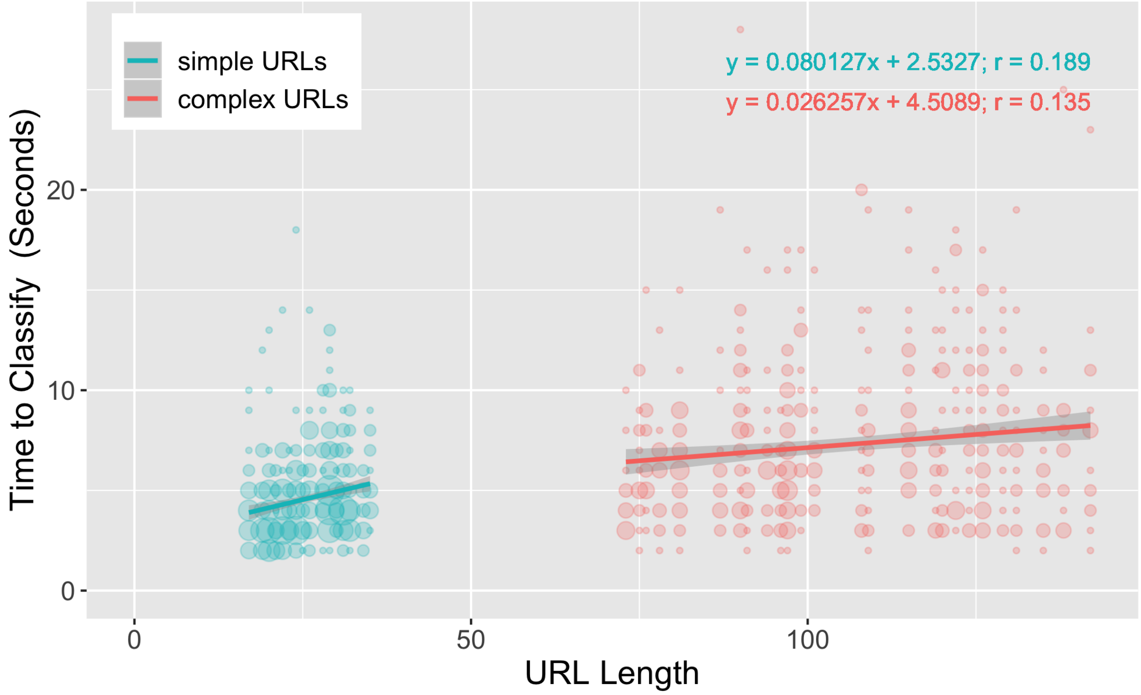


Figure 5.4: Time spent to classify URL vs. URL length with linear regression lines for *simple* and *complex* URLs. As URL length increases, the time participants took to classify the URLs also increased. However, the rate of increase is much smaller for complex URLs in comparison to simple URLs.

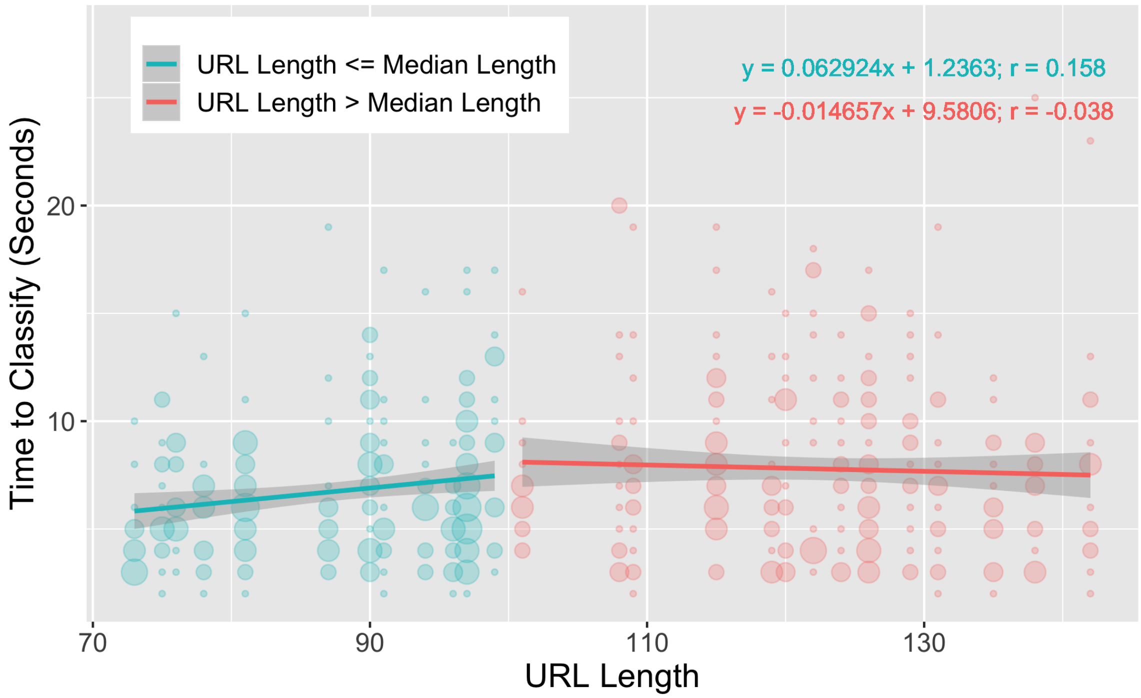


Figure 5.5: Time spent to classify URLs vs. URL length with two linear regression lines for data points separated by the median URL length (*complex* URLs). This graph suggests that there may be a peak URL length after which increasing URL length and complexity does not lead to any more time spent on classifying the URL.

suggests a peak in time spent per character that occurs near 100 characters. We observed similar trends with fixation count per character and backtrack fixation count as a function of URL length for *complex* URLs.

5.5.5. Existence of *www*

We compared *safe* URLs that have `authority` components that begin with *www* (C1&C3) to those that do not (C2&C4). Wilcoxon test results indicated a significant difference in time spent per character on the ***authority AOI*** between *www* URLs (M=0.16, SD=0.04) and *non-www* URLs (M=0.21, SD=0.04); [Z=4.2094, p<0.0001]. Also, there was a significant difference in the fixation count per character on the ***authority AOI*** between *www* URLs (M=0.24, SD=0.09) and *non-www* URLs (M=0.34, SD=0.12); [Wilcoxon test: Z=3.2292, p=0.0012]. The score obtained

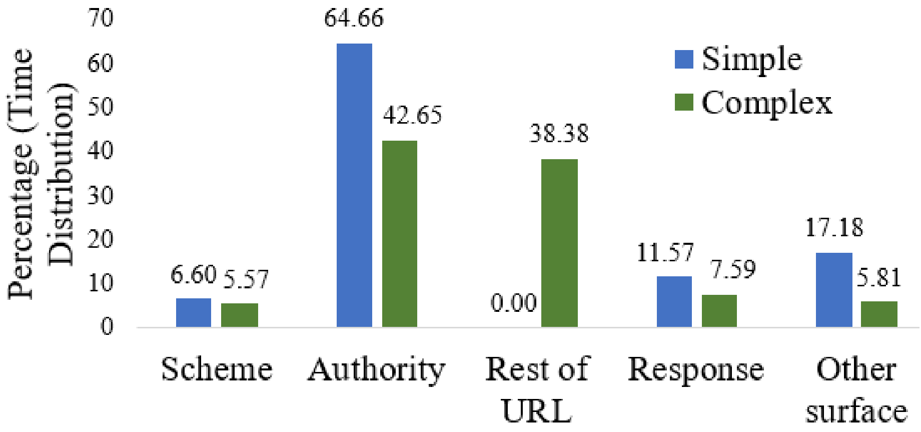


Figure 5.6: Percentage of Classification Time spent on the AOIs. This figures shows that for both *simple* and *complex* URLs, users spent the least time on the `scheme` component (note the `rest` doesn't exist for simple URLs) and the most time on the `authority` component on average.

(maximum score: 8) was also significantly different between *www* URLs (M=6.50, SD=1.48) and *non-www* URLs (M=4.09, SD=1.90); [Wilcoxon test: Z=4.7020, p<0.001].

5.5.6. User@Host Attack Type vs. Regular URLs

To examine user visual attention for the *user@host* URLs (C8), we considered two special AOIs at a finer granularity than the ***authority AOI***: the ***user AOI*** and ***host AOI***. We compared measurements on these two AOIs for the *user@host* URLs (C8) to those for the ***authority AOI*** for safe URLs of similar structure (C3). Using the Kruskal-Wallis test we found a significant difference on time spent per character between the ***authority AOI*** of C3, the ***user AOI*** of C8, and the ***host AOI*** of C8 [$X^2(2)$=32.1735, p<0.0001]. A significant difference was also observed with fixation count per character [Kruskal-Wallis test: $X^2(2)$=11.3323, p=0.0035]. Post hoc analysis indicated that both sets of measurements for the ***host AOI*** for C8 were lower than those of the ***user AOI*** for C8 and the ***authority AOI*** for C3; the measurements between the ***user AOI*** for C8 were comparable to those of the ***authority AOI*** for C3. These results suggests that users process the ***user AOI*** of

Hypothesis	*Result*
H1: Total time spent on processing a URL is longer for complex URLs than it is for simple URLs.	True
H2: Total time spent on processing a URL, normalized by the URL length, is shorter for complex URLs than it is for simple URLs.	True
H3: There exists a URL length threshold over which increasing URL length does not result in more time being spent on processing URLs.	True
H4: Total time spent on the `scheme` component per character is less than that of the `authority` and `rest` components.	False
H5: For URLs that have an `authority` component of form `user@host` where `user` ends with ".com", participants spend significantly more time per character looking at the `user` component than the `host` component.	True

Table 5.6: Results of our hypotheses. This table explains which hypotheses we found evidence to support.

C8 and the ***authority AOI*** of C3 similarly. Also, there was a significant difference in the score between the user@host attack type (M=3.37, SD=2.41) and safe URLs of similar structure (M=5.81, SD=1.51); [Wilcoxon test: Z=2.9176, p=0.0035].

Section 5.6

Discussion

Participant responses to the pre-task questionnaire following the mood induction video [167] indicated they were awake and in a neutral mood. Responses to the post-task questionnaire reveal that participants did not fatigue, and, on average, correctly identified the safety of about 40 of the 64 URLs (63%).

We now turn to a detailed discussion of the results. Table 5.6 specifies which hypotheses are supported by our results.

5.6.1. URL Processing & Classification Factors

URL Length:. The overall time spent on classifying *simple* (and shorter) URLs (C1, C2, C5, C6) was less than the total time spent on classifying *complex* (and longer) URLs (C3, C4, C7, C8). This weakly supports ***H1***, though follow-up work must be done to disentangle length from other complexity factors.

For *complex* URLs, we found URL length negatively correlated with time spent per character and fixation count per character. This supports ***H2***.

We did not observe a correlation between URL length and score. Also, while Figure 5.4 suggests participants spent more time parsing URLs as URL length increases, Figure 5.2 suggests time spent per character decreases as we increase URL length. Moreover, the positive correlation between URL length and time spent seems to cease at a point, which supports ***H3***. Specifically, Figure 5.5 suggests that at a threshold of approximately 100 characters, time spent stops increasing as we increase URL length. Similar trends were observed with fixation count per character and backtrack fixation count per character. We also observed no statistical difference between time spent on complex URLs under 100 characters and those above. One interpretation is captured by a notion similar to that of the compliance budget proposed by Beautement et al. [20]: the user may only expend a finite budget of resources (here, time is a proxy for expended resources) to classify a URL, and, if the resources required to fully process a URL exceeds this budget, the user will not expend them. While the peculiarities of where that threshold is may depend on factors other than just URL length, we expect this notion of a finite budget applies more generally.

AOI:. We examine the influence of the AOIs:

- **Scheme AOI**: The decrease in the pupil area for the ***scheme AOI*** indicates reduced cognitive attention. Previous work found the frequency with which a user encounters a word affects the fixation duration and processing of that word [155]. Users usually spend less time on frequently encountered words. Most legitimate websites use *https* nowadays, which is also used in each of the 64 URLs in our corpus. This explains the decrease in cognitive load for the ***scheme AOI***. We observed a statistically significant difference in time spent per character between the ***scheme AOI*** and the ***authority AOI*** (with the latter being higher); however, we did not observe such a difference for the ***scheme AOI*** and the ***rest AOI***. Therefore, we do not have evidence to support ***H4***.

- **Authority AOI:** The results indicate the time spent per character on the ***authority AOI*** is significantly higher than that of other AOIs. Time spent and fixation count per character on the ***authority AOI*** suggests users find *www* at the beginning of the domain name to be a strong indicator of URL safety.

- **Rest AOI:** Reduced fixation count while reading is characteristic of scanning text [156]. The fixation count per character for the rest AOI is significantly lower than it is for other AOIs, which suggests participants scanned the ***rest AOI***.

Attack Types:. Participants classified *positive*, *unsafe* URLs (C5) correctly 55% of the time and they classified *negative*, *unsafe* URLs (C6) correctly 74% of the time. This suggests people are more inclined to trust URLs that use positive words or phrases, even if they have no familiarity with the domain name. Table 5.4 shows that participants, on average, correctly classified the URLs 77% of the time, given that they had heard of the associated services.

Results suggest users visually process the `user` component of URLs with the *user@host* attack type (C8) similar to how they process the `authority` of URLs without a `user` component. In general, the fixation count per character was low for the `rest` component relative to both the `scheme` and `authority` components. For C8, we observed a reduced fixation count per character and time spent per character on the `host` component, which suggests participants perceived the `host` component as part of the `rest` component. Visual evidence suggests participants misidentified the `user` component as the `host` for URLs in C8. Of the *unsafe* URL categories, participants scored worst on C8. Participants spent significantly more time per character on the `user` component than the `host` component for C8, in support of ***H5***.

We expect classification accuracies observed in this study are upper bounds on what users achieve in practice without additional safeguards in place. Sophisticated attacks that use URL features participants do not know about will likely be more effective. We also expect that attacks that use obfuscation in the `rest` component—or rather, what users *perceive* as the `rest` component—are more likely to succeed given that participants spent less time on the `rest` component than the `authority` component in our study.

5.6.2. Improving Security in Practice

The study suggests a sort of ceiling effect: as URL length increases, participants spent more time vetting the URL until it capped out at around 100 characters. It also provides visual evidence of user misperceptions regarding URL structure. These insights into how users process and perceive URLs suggest concrete steps and best practices for services to improve the perceived security—and, we argue, the *actual* security—associated with the URLs they serve. For example, from a purely technical standpoint, there is no intrinsic security benefit to serving a URL that is short, has a domain name that begins with *www*, and has few special characters. But if those URLs match users' safety expectations, users would be better at classifying both safe URLs served by the service and unsafe, obfuscated URLs served by adversaries.

Some *unsafe* URLs from our corpus were classified as safe because they exploited uncommon URL features that users rarely encounter in practice with legitimate services. Ironically, this makes such URLs easy for a computer to classify as risky. Surprisingly, we found that some web browsers offer no user protection against such URLs, even though simple-to-write parsers could easily detect them. This provides an opportunity to improve security at minimal cost.

Last, our findings can improve the quality of security awareness training programs. Our study identifies various misperceptions held by users. It also provides concrete evidence of where users look as they process URLs. This study's methods and data may help in assessing, comparing, and improving training modules that aim to help users correctly identify URLs.

Section 5.7
Limitations

Several considerations may have affected study generalizability: Participants were predominantly male college students pursuing electrical engineering degrees. To ensure the eye tracker accurately picked up on AOIs, we used a large font and displayed URLs over multiple lines. URLs were presented in isolation; contextual factors (e.g., the device on which a URL is displayed, the application on which a URL is viewed, or beliefs regarding who sent it) may affect visual behaviors and responses. Also, repeatedly asking participants whether URLs were safe likely sensitized them to phishing attacks.

However, we took precautions to minimize unintended effects. We conducted pilot runs to ensure the interface was clear and user fatigue was minimized. We used the post-experiment questionnaire to evaluate experimental validity. And we used a neutral-mood-inducing video to reduce variability in mood.

The available indicators provide some evidence of the study's validity. The average participant score of 63% is within the ballpark of similar studies, e.g., [52, 173].

Post-task survey responses indicate most participants took the task seriously, exercised equal or only slightly more caution than they would in practice, and were not fatigued. Although no data we collected suggests a significant bias, we expect that the artificiality of the experimental context, wherein users classified URLs in series, would have had some effect on the the classifications and visual processes. That said, we believe any bias would be in the direction of more caution and would be unlikely to invalidate our security recommendations as problems during the classification task would continue to be problems in the real world. We also note that applications and interfaces in the wild may vary regarding font properties so there is no one-size-fits-all approach for conducting such studies.

Last, the URLs may have had features we could not identify that affected participants' visual behaviors and responses. We attempted to mitigate these concerns by including eight URLs per category, but further work is needed. Also, we only considered a few flavors of URL-based attacks. Notably, no attacks made use of the `rest` component, which may have affected participants' visual behaviors.

Section 5.8

Conclusion and Future work

Eye tracking is a lens through which we can keenly understand user security behavior. The work presented in this chapter is a first step toward developing a model that captures how users visually process, derive meaning from, and operationalize URL security information to gauge URL safety. We conducted a user study in which participants saw URLs and then classified them while wearing an eye tracker. The findings suggest that participants relied on poor security indicators such as presence of *www* to gauge URL legitimacy, that they spent more time and cognitive resources to vet longer URLs but only up to a point, and that, for the unsafe, *user@host* URLs, participants perceived the `user` component to be the `host` component. In future work, we plan to study other contextual factors such as mood, additional flavors of URL obfuscation, and the effectiveness of training the user.

Chapter 6

An MTurk Study Examining How Users Evaluate URLs

In the previous chapter, we used eye tracking to explore the visual behaviors of users while they parsed and classified URLs. Here, we run a complementary, large-scale study with two primary aims: to determine the unsafe URL structures to which users are most susceptible and to examine how factors, including URL features and font, affect users' assessments of URL safety. We recruit participants over Amazon Mechanical Turk to take part in the study, which, again, involves classifying URLs; we record user responses and the time taken to classify each URL. Although this approach does not provide data on how users visually process URLs, the reach of MTurk allows us to study a much larger population of users. This means we can achieve the requisite sample size to detect smaller phenomena, that we can use a larger URL corpus, and that we can look at a variety conditions.

Section 6.1

Introduction

How users perceive the security information presented to them impacts their security behaviors and, in turn, the state of security at both the individual level and the organizational level. Thus, it is paramount that we have a clear conception of how users interpret security information. In the previous chapter, we sought to learn how users process URLs from a visual standpoint. Participants classified URLs while wearing an eye tracker, which provided reliable, ground-truth information on the visual processes employed during URL processing. Additionally, the link between pupil dilation and cognitive load allowed us to draw strong inferences about underlying cognitive processes. However, the shortcoming of such studies is that they require a lab with sophisticated equipment. This requirement, coupled with the other requirement of a significant time investment on both the part of researchers and participants, severely limits the number of participants one can have and, therefore, the types and number of research questions that can be pursued.

The work presented in this chapter complements our eye-tracking work. We conduct a large-scale study to understand how people classify URLs using the platform Amazon Mechanical Turk. While we do not get the fine-grained visual data that tells us where people look as they process URLs, what parts they pay attention to, and where they struggle, this study has the benefit of a significantly larger population of users. This, in turn, enables us to examine many different classes of URLs, test hypotheses corresponding to relatively subtle phenomena, and explore a variety of conditions that affect how users evaluate the safety of URLs.

Section 6.2

Background

In this section, we discuss related work and present the requisite terminology to understand our work.

6.2.1. Related Work

In the interest of avoiding excruciating redundancy in this thesis, the related work presented here is relatively compact. We only cover the papers which are most closely aligned with the work in this chapter. The related work presented in Section 5.2 surveys the broader phishing literature. Similarly, we assume the reader is familiar with URLs and URL structure. If not, we suggest they consult Section 5.2.5 of this thesis and/or relevant specifications and standards [25, 26, 198].

Many different types of studies have been conducted to understand different aspects of URL security. Researchers have used machine learning to automatically classify URLs, e.g., [187]. Training systems and assistive technologies to aid users have also been developed. For example, Althobaiti et al. [9] reported on results from a study on the efficacy of a Slack chat bot designed to assist users in assessing URL safety. For another pair of examples, Conva et al. [43] and Sheng et al. [172] developed games to train users to not fall for phishing URLs. A variety of studies have examined phishing and URL obfuscation attacks as they pertain to emails and websites, e.g., [52, 79, 173]. However, there are some key differences between our study and many of these phishing studies. As far as we can tell, in comparison to many of these earlier studies, the project described in this chapter is wider in scope with regard to the number of URLs studied. And we examine URL classification without the presence of accompanying context, which has trade-offs. Notably, additional contextual information does weigh into users' security evaluations; however, it also introduces additional factors which make it hard to tease out what role the URL itself had on the classification. Some work examines the impact of emotion on phishing susceptibility, e.g., [186]. In our study we examine the impact that the *valences* (positive/neutral/negative) of words that make up the fully qualified domain name of URLs have on how users perceive those URLs.

Quinkert et al. [149] conducted a large-scale study on Amazon Mechanical Turk where participants classified URLs and also constructed mock phishing URLs for others to classify; they found this process was effective in training users to detect malicious URLs, but it also produced false negatives in instances where legitimate URLs had features that were similar to illegitimate ones. Albakry et al. [6] also conducted a large-scale study on Amazon Mechanical Turk, which sought to examine whether people could identify where a URL would take them, as well as whether they felt URLs would be safe to visit. The paper has many similarities to our work. Indeed, it is the most closely related work that we saw. However, it also has many key differences. Our aims are slightly different: Whereas the study by Alkabry et al. focused more on determining whether users could identify where URLs would take them, we solely focus on understanding whether users think a URL is safe to visit (user notions of safety and the labels we assign to URLs will be discussed soon!). Our URL corpus is significantly different from theirs in size and the URL structures studied, though there are some overlaps. We also examine the impact that the font makes on responses. Additionally, we look into the duration of time taken to classify a URL.

6.2.2. Some Basics

This study examines user susceptibility to many URL structures. While we shall define URL features when we present our corpus, we present some key terminology and ideas that will help in understanding the discussions motivating how we went about constructing our corpus, as well as the URL features themselves:

- *URL redirection* is a technique that lets a service automatically redirect the user who visits a URL X under the service's control to some other URL Y, either with or without user interaction after visiting X. [203] URL redirection has legitimate uses, but it also can be abused. Many legitimate services embed the URL to be directed to (Y) within the redirector URL itself (X) and then employ a vetting process to ensure that the URL to be redirected to (Y) is safe, for some definition of safe, before a seamless transition takes place; in the event that the URL to be directed to (Y) is determined to be unsafe, the service may warn the user that a redirection is about to take place as a safety precaution, and they may require user consent before the redirection takes places. However, some services do not provide such a check or that check can be bypassed by craftily constructing URLs. URLs susceptible to these attacks are called open redirectors. Open redirectors are often exploited to conduct phishing or other attacks. (E.g., ***https://business.facebook.com/ads/creativehub/select/?redirect_uri=https%3A%2F%2Fbit.ly/p5wv65V***)

- URL shorteners [212] are one interesting class of URL redirectors. URL shortening services allow users to construct short URLs that redirect to longer ones. However, this usually means that users cannot glean much information about the shortened URL by simply looking at it. (E.g., ***https://bit.ly/0B3GQ1***)

- When we talk about *gatekeeper* URLs, we are talking about URLs created by gatekeeper security services such as Microsoft Office 365 APT Safelinks [118] and Proofpoint URLDefense [148]. These gatekeeper services act as an intermediary, rewriting URLs sent by email so that the recipient is presented not with the URL that the sender sent, but rather the rewritten URL. The rewritten URL redirects to the actual URL that the sender sent if it satisfies the vetting process that the gatekeeper service has in place. We also note that the actual URL to which a service is being redirected to is embedded in the gatekeeper URL. However, even with significant interaction with gatekeeper services, it is unlikely users can correctly interpret the embedded URL information, if it exists at all. Indeed, this is a topic of exploration in our study. (E.g., ***https://nam01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.youtube.com&dat*** ...)

- *Homograph* attacks exploit user perceptions of where a URL goes by substituting a character string for another one that looks awfully similar. While there are more sophisticated IDN homograph attacks, we study user susceptibility to the more basic ASCII variant, as well as how font affects susceptibility. [206] (E.g., ***https://www.zilIow.com***)

- We compare two fonts in this study: a regular font that is representative of the font style users are shown in the wild when assessing URL safety and a monospaced font [208] for which every character is the same width. We compare these fonts as the choice of font may affect user susceptibility to *homograph* attacks.

- The valence of a word or phrase is a measure of how positive/negative a word is. [121] We examine the impact of the valence of the words contained within the fully qualified domain name of a URL on user classification for a subcorpus of relatively short URLs. (E.g., ***https://www.farm-living.com*** , ***https://www.datageek.com***, ***https://www.furydemolition.com***)

Section 6.3

Our Aims and Contributions

We aim to discover how users process security information, specifically the information embedded in URLs. We seek to learn which URL structures are most effective at deceiving users and to discover what factors affect the safety decisions users ultimately make. The end goal of this research thrust is to deliver data that improves aggregate security in practice, e.g., by suggesting techniques that improve security training or by providing tools that can be incorporated into existing software to improve users' mental models of URL safety.

More specifically, we seek to deliver—and we believe we do deliver—the following contributions:

- We examine user susceptibility to some newer URL structures and attacks, including gatekeeper URLs [118, 148] and URL shorteners. Both have been looked at by Althobaiti et al. [9]; however, that study's focus was more on assessing the quality of a defense mechanism than how users respond to specific URLs, only one URL was studied per category and only 20 participants belonged to a condition. To the best of our knowledge, user perceptions of gatekeeper URLs have not been examined before in usability studies. We note that Albakry et al. [6] has examined how users evaluate URL shorteners. We additionally examine the impact that the valence [121] of the words in the domain name have on phishing susceptibility.

- We examine the impact of font style on user susceptibility to phishing attacks. In particular, we compare how users respond to unsafe homograph attacks using a regular font and a monospaced font.

- Previous studies have examined a variety of URL obfuscation attacks and this study re-examines some of them. Re-evaluating old findings is critical in science but it is especially necessary when those findings pertain to technology and usability as the way users interaction and understand technology is constantly evolving. Over the years, both the kinds of URLs that people interact with and the ways that users interact with them have changed. For just one example, a decade ago, many popular services did not use HTTPS, whereas the vast majority of services support it today. Many URL obfuscation techniques and features we examine in this work have been studied before in the literature. However, we study URLs in isolation, as opposed to other studies where URLs are studied in relation to, say, an email. While studying URLs in emails provide contextual factors that do occur in practice, they often add confounding variables in presentation that are hard to account for. In addition to the general value as a replicability study, we study a relatively large subpopulation of users and four URLs per URL class instead of just one, enabling us to get a better handle on which attacks succeed and fail compared to many other studies of this kind.

Section 6.4
Hypotheses

Our full hypotheses will explore two measures: the URL classifications themselves and the time taken to make those classifications. The rationale for studying the former measure is obvious. The rationale for studying the latter measure is that it tells us something about how a user processes a URL. We expect the duration of time associated with URL classification to positively correlate with the amount of time users spend fixating on parts of the URL. As we mentioned in the previous chapter, users extract information when they fixate on text. A long fixation duration indicates a high cognitive load [90, 204]. If we have a long URL and a short classification time, for example, this may suggest that participants are only scanning, not reading, parts of the URL and that they struggle to derive meaning from the URL, resulting in them giving up on the classification process early. Of course, to get a clear picture of what is going on, it's useful to have more information. But time taken to classify URLs is, nonetheless, a useful indicator of what kinds of URLs people feel they can interpret and whether they expend the effort to try to interpret them. That said, we are only presenting preliminary results here and do not examine any hypotheses related to time taken to classify URLs. We have defined the other hypotheses before we conducted this experiment, and we will report on them in other published work. However, they are not discussed here.

***H1*:** Participants are better at classifying safe short URLs than they are at classifying long URLs.

***H2*:** Participants respond to safe and unsafe gatekeeper URLs differently.

***H3*:** Participants classify unsafe positive-valence URLs as safe more than they do unsafe negative-valence URLs.

***H4*:** Participants more accurately identify unsafe URLs in monospaced font than they do unsafe URLs in a regular font.

Section 6.5

Corpus

We designed our URL corpus and method in pursuit of testing our hypotheses. We then iteratively revised the corpus and hypotheses until we felt satisfied that testing the hypotheses would provide a valuable contribution to the field and that the corpus we created would allow us to test those hypotheses. We also reflected on and revised the experimental method during this process.

In this section we explain and justify how we applied the safe and unsafe labels to URLs, we discuss how we went about creating the URL corpus, and we enumerate the URL categories we ultimately selected.

6.5.1. What is a Safe URL?

This study has two notions of safety associated with URLs. The first is what users think of as safe URLs. This corresponds to the responses participants give us when we ask them whether a URL is safe. The second, which we discuss in this subsection, corresponds to the safe and unsafe labels that we assign to URLs for the purpose of analysis. This sets the baseline for what is a correct or incorrect user classification, which we use to test our hypotheses. In this subsection, we discuss how we apply those labels, and we provide the rationale for the labeling.

We label a URL as safe if either (1) its domain name belongs to a legitimate service and it does not involve redirection or (2) its domain name belongs to a legitimate service that redirects to a safe URL. Of course, this raises the question of what it means for a domain name to belong to a legitimate service. We mean that the domain name is either routinely visited by a large fraction of the general population for the purpose of receiving a service (e.g., news, banking, social networking) or it is a subdomain of such a domain.

Admittedly, this definition is imperfect. Given the ambiguity and subjectivity inherent in notions of safety, any strict rule-based definition will have flaws. Below, we present some critiques of our definition and respond to them with the intent of communicating the rationale for our definition.

- Some may argue that it is illogical to assign safety to a URL based solely on its character string. After all, what about URL hijacking, man-in-the-middle attacks, and the reality that the domain registry changes? We agree that such considerations are necessary in assessing any absolute notion of safety. However, important security information is also conveyed through the URL string itself. Moreover, in many cases, for many people, the URL is the only security information they are presented with before they make a decision of whether or not to visit a website. It is imperative that we get a handle on how these decisions are made.

- Another concern is that safety is not a binary attribute and some individuals do not perceive it as binary. It is true that in some circumstances, some users may be unsure of the safety of a URL and take additional precautionary steps to vet the URL. However, given the sheer number of URLs that users are routinely presented with, our findings from the eye-tracking study presented in the previous chapter, and the reality that even in situations where users take precautionary steps, there is still a binary decision of whether to immediately click on the URL or not. Thus, it seems appropriate to present users with a binary choice.

- Last, the definition is somewhat imprecise. While true, this definition serves as a guiding notion that drives us toward a more concrete URL selection approach we shall soon explain.

Above, we explained what we consider to be a safe URL. We label a URL as unsafe if any of the following applied at the time of URL corpus construction:

- The URL was eligible for purchase.

- We could not find a DNS record corresponding to the URL.

- The URL was a spoof of a legitimate URL not claimed by the target service (we note that one such URL corresponded to a legitimate service that claimed the URL of another legitimate service to demonstrate what a homograph attack looks; it's exceedingly unlikely participants knew this *a priori*).
- The URL redirected to an unsafe URL. We also used open redirectors from the recent past that were no longer valid at time of corpus construction.

Although visiting some URLs labeled as unsafe may not pose a security risk in practice, these extra allowances allow us to construct URLs in a more structured fashion, which in turn allows us to obtain a better understanding of how users process URLs. Additionally, we do not believe the user would have the requisite information to come to the determination that such URLs are safe in the experiment by just seeing them; that is; it is exceedingly unlikely that users would have visited such URLs before. Thus, users truly should be classifying them as unsafe.

Some URLs, importantly unpopular and only moderately URLs that correspond to legitimate services, do not fall under either the safe or unsafe categories mentioned above. This is intentional. Notably, for safe URLs, we wanted only to explore those URLs that corresponded to services that most people would have interacted with. While a user who has not interacted with a URL offered by a legitimate service may classify it as unsafe, we believe such a user would be making the correct choice.

6.5.2. Constructing the URL Corpus

With the notions of safety we outlined in the previous subsection, we now present how we went about creating the URL corpus.

To create the subcorpus of safe URLs, we relied on the "The top 500 sites on the web" provided by Alexa on March 14, 2020 for USA, which ranks site popularity based on web traffic.[1] We used a subset of these websites with the intent of creating different classes of safe URLs where the average rank of the URLs in each class are roughly the same. We also tried to stick to websites that we believe many users will have heard of and that we believe are only visited intentionally by users. We created URLs by using the domain names from the list prepended by *https://* and by finding URLs that were only a couple of clicks from these main pages, which we expect users to visit and share, that share the same top-level domain and second-level domain as the main site. Additionally, we used gatekeeper services and URL-shortening services to create URLs based on these domains. To generate gatekeeper URLs, we learned the structure of the URLs provided by the gatekeeper service, substituted query values with random strings chosen from what we believe are similar distributions of query values for legitimate URLs, used an embedded URL representation of the site we were redirecting to, chose a randomized embedded email address of form *FirstName.LastName.FiveDigits@gmail.com* using DuckDuckGo's randomization functions if needed, and embedded the redirected URL in its appropriate spot, modified in the ways the gatekeeper services would. The shortened URLs were created by choosing random characters for the strings at the end of these URLs.

To create the subcorpus of unsafe URLs, we first created a draft of URL classes by consulting corpora of existing malicious and phishing URLs and domains [137, 113, 162, 169]. To create the corpus itself, we used a mixture of real phishing websites from those phishing corpora [137, 113, 162, 169], URLs used in recent phishing attacks from blog posts and bug bounty writeups [195, 3, 126, 192], and URLs we created that were eligible for purchase or did not have a DNS record associated with them. While one can argue that some of these URLs may not be unsafe even though they are labeled as such, the users are extremely unlikely to have heard of those URLs,

[1] Available from: https://www.alexa.com/topsites

and, therefore, they *should* be classified as unsafe by the user. We reiterate that the purpose of not solely using existing phishing attacks is that it enables us to have greater control over the structure and properties of different URL classes, facilitating a more reliable analysis.

6.5.3. URL Features

We discuss the URL features. For some of the features, we only considered their applicability within a small subcorpus of URLs, but ignored them outside of that subcorpus. Thus, we do not list all the features that apply to each URL class, only those that were applicable for analysis.

Dummy:. The *dummy* feature corresponds to those URLs which were simply placed at the beginning of the URL classification task to address early learning effects that participants experienced as they became accustomed to the interface. No data about these URLs was used in analysis.

Canary:. The *canary* feature is associated with canary URLs that we used to remove outliers. These corresponded to extremely popular services, specifically those in the top 4 of the Alexa rankings, corresponding to the services Google, YouTube, Facebook, and Amazon. If a user classified a *canary* URL as unsafe, we discarded all data from that participant for the purpose of analysis.

Safety:. As discussed earlier, we assign the labels of *safe* and *unsafe* to URLs. Additionally, we used the *unknown* label for URL shortener URLs.

URL Length:. We considered the following features, which capture the number of characters within the URL:

- *short*: ≤ 32 characters

- *medium*: 33-64 characters
- *long*: 65-96 characters
- *very long*: 97-128 characters
- *extremely long*: >128 characters

Top-level domain is* `com`*:. The *com* feature is applied to URLs that have a top-level domain of `com`, i.e., the domain name ends in `com`. The *non-com* feature is applied to those that do not.

Bottom-level domain is* `www`*:. The *www* feature is applied to URLs that have a bottom-level domain of `www`, i.e., the domain name begins with `www`. The *non-www* feature is applied to those that do not.

Existence of Path:. The *path* feature is applied to URLs that have an explicit path following the fully qualified domain name, which is not just `/`. The *non-path* feature is applied to those URLs that end with the full qualified domain name and perhaps one `/` character thereafter.

Font:. The two fonts are *regular* which corresponds to the font Arial and *monospaced* which corresponds to the font Go Mono.

Gatekeeper:. Within our study, we considered gatekeeper URLs, which had the *gatekeeper* feature applied to them; they corresponded to both safe and unsafe URLs. We wanted to see how much effort users expend in parsing these URLs. We examined URLs served by both Microsoft Office 365 APT Safelinks [118] and Proofpoint URLDefense [148].

Levels of Fully Qualified Domain:. For *short*, *safe*, *non-www*, *non-path* URLs, we looked at the impact of the level of domains in the fully qualified domain name. Those with two levels had the *two domain levels* tag applied; those with three had the *three domain levels* tag applied.

Valence:. The valence of the URL is derived from the valences of the constituent words in the fully qualified domain name in accordance with NRC VAD Lexicon [121]. We only explore features related to valence for a small subcorpus of URLs of similar form to determine the impact of valence on phishing susceptibility of unsafe URLs. The relevant features are:

- *positive*: The fully qualified domain name comprises two words, possibly hyphenated, each of valence ≥ 0.8.
- *neutral*: The fully qualified domain name comprises two words, possibly hyphenated, each of valence within the range $0.4 - 0.6$.
- *negative*: The fully qualified domain name comprises two words, possibly hyphenated, each of valence ≤ 0.2.

Attack Techniques:. In addition to the valence features, we considered a number of attack techniques for *unsafe* URLs:

- *homograph*: This feature is applied to URLs that use the ASCII homograph [206] attack technique. Two of the URLs used an uppercase *i* in place of a lowercase *l*; two used *rn* in place of *m*. (E.g., *https://www.zillow.com* → *https://www.zilIow.com*)
- *combosquatting*: This feature is applied to URLs that start with a substring of the domain that is legitimate but then add text within one domain level. (E.g., *https://www.adobe.com* → *https://www.adobe-update.com*)

- *infix domain*: This feature is applied to URLs where we add a domain label in the middle of the fully qualified domain name. (E.g., `https://www.youtube.com` → `https://www.youtube.yt-red.com`)

- *wrong TLD*: This feature is applied to URLs that take the form of a legitimate URL but with the top-level domain swapped with something else. (E.g., `https://www.spotify.com` → `https://www.spotify.vg/`)

- *domain-in-domain*: This feature corresponds to URLs that have a fully qualified domain name that contains an unaltered safe fully qualified domain name, followed by more domain levels that make it unsafe. (E.g., `https://www.att.com` → `https://www.att.com.att-wl.com`)

- *redirector*: This feature corresponds to open redirectors. The domain corresponds to a legitimate service, but the URL suffers from an open redirect vulnerability that has been exploited in the past (see: [195, 3, 126, 192]). Many, perhaps all, of these attacks no longer work. However, they have been used in the recent past.

- *hex obfuscation*: Characters within a URL can be represented using hexadecimal notation, e.g., `%2E` maps to the `.` character. This feature was applied to obfuscate the destination of the URL. (E.g., `https://www.imdb.com` → `https://www.imdb.com@imdb-go%2E%63%6F%6D`)

URL Shorteners:. Last, we considered URLs served by URL shortening services [212]. These URLs had the *url shortener* feature applied.

Section 6.6

URL Classes

Our URL corpus comprised 116 URLs in total. While there are many ways to cut our corpus into classes depending on what hypothesis is being explored, we present the partition of classes we used the most in Table 6.1. Each of these classes comprised 4 URLs.

Some quick notes:

- In some cases, a class feature varied across class members. We either used *mixed* for the feature to indicate this or we used a special tag to indicate what subset of features were applied. We use *short-very long* to indicate that the class contains one instance each of a *short* URL, *medium* URL, *long* URL, and *very long* URL. We use *medium-very long* to indicate that the class contains URLs that have lengths between *medium* and *very long*. Similarly, we use *long-very long* to indicate that the class contains URLs that have lengths between *long* and *very long*, respectively.
- For every URL represented by these classes, we technically had two URL images, one corresponding to the *regular* and *monospaced* font features. However, in the interest of not having a table of twice the size, we leave out this feature. In the results and analysis, the font we are using in comparisons should be clear.

Section 6.7

Method

In this section, we provide an overview of the experiment, explain the task in further detail, state how we selected the participants, and finally, we explain what we

Class	**safety**	**length**	**com**	**path**	**other**
C1	*mixed*	*mixed*	*mixed*	*mixed*	*dummy*
C2	*safe*	*short*	*com*	*non-path*	*canary*
C3	*safe*	*short*	*com*	*non-path*	*www*
C4	*safe*	*short*	*com*	*path*	*www*
C5	*safe*	*short*	*com*	*non-path*	*non-www, two domain levels*
C6	*safe*	*short*	*com*	*non-path*	*non-www, three domain levels*
C7	*safe*	*medium*	*com*	*path*	
C8	*safe*	*long*	*com*	*path*	
C9	*safe*	*very long*	*com*	*path*	
C10	*safe*	*extremely long*	*com*	*path*	
C11	*safe*	*medium*	*non-com*	*non-path*	
C12	*safe*	*long*	*non-com*	*path*	
C13	*safe*	*very long*	*non-com*	*path*	
C14	*safe*	*extremely long*	*non-com*	*path*	
C15	*safe*	*extremely long*	*com*	*mixed*	*gatekeeper*
C16	*unsafe*	*short-very long*	*com*	*mixed*	*homograph*
C17	*unsafe*	*short-very long*	*com*	*mixed*	*combosquatting*
C18	*unsafe*	*short-very long*	*com*	*mixed*	*infix domain*
C19	*unsafe*	*short-very long*	*non-com*	*mixed*	*wrong TLD*
C20	*unsafe*	*short-very long*	*com*	*mixed*	*domain-in-domain*
C21	*unsafe*	*medium-very long*	*com*	*mixed*	*domain-in-domain, hex obfuscation*
C22	*unsafe*	*short-very long*	*com*	*mixed*	*user@host*
C23	*unsafe*	*medium-very long*	*com*	*mixed*	*user@host, hex obfuscation*
C24	*unsafe*	*short*	*com*	*non-path*	*www, positive*
C25	*unsafe*	*short*	*com*	*non-path*	*www, neutral*
C26	*unsafe*	*short*	*com*	*non-path*	*www, negative*
C27	*unsafe*	*long-very long*	*com*	*path*	*redirector*
C28	*unsafe*	*extremely long*	*com*	*path*	*gatekeeper*
C29	*unknown*	*short*	*com*	*path*	*url shortener*

Table 6.1: A summary of the 29 URL classes.

measured and how we did the analysis.

6.7.1. Overview

Participants were recruited over Amazon Mechanical Turk between March 15, 2020 and March 16, 2020 to take part in a *Human Intelligence Task (HIT)*—a (usually short) task or job to be performed on Amazon Mechanical Turk for payment. The URL classification task involved classifying URLs and filling in a post-task questionnaire. We received an IRB exemption under Title 45, Subtitle A, Subchapter A, Part 46, Section 104, Category 2 of the Code of Federal Regulations.[2] After accepting the task on Amazon Mechanical Turk, the participant was directed to the platform Qualtrics [209] to perform the task. Upon completing the task on Qualtrics they were given a unique code to enter onto Amazon Mechanical Turk as proof of completion.

6.7.2. HIT Details

The HIT could conceptually be divided into two parts: a URL classification task and a post-task questionnaire. Technically, the whole task was part of a single questionnaire broken up into different blocks on Qualtrics. The questionnaire included the following components in the order presented:

- First, participants were shown an informed consent sheet that provided a very brief summary of the experimental aims, the data we were collecting, how we planned to use the data, the expected time for completion, and other things one would expect on such a sheet. Participants knew they were taking part in a task that involved classifying URLs and they knew the task contained a post-task questionnaire. Beyond that, they were not privy to experimental details such as the group they were randomly being assigned to.

[2] An electronically accessible version is available at `https://www.ecfr.gov/cgi-bin/ECFR?page=browse`.

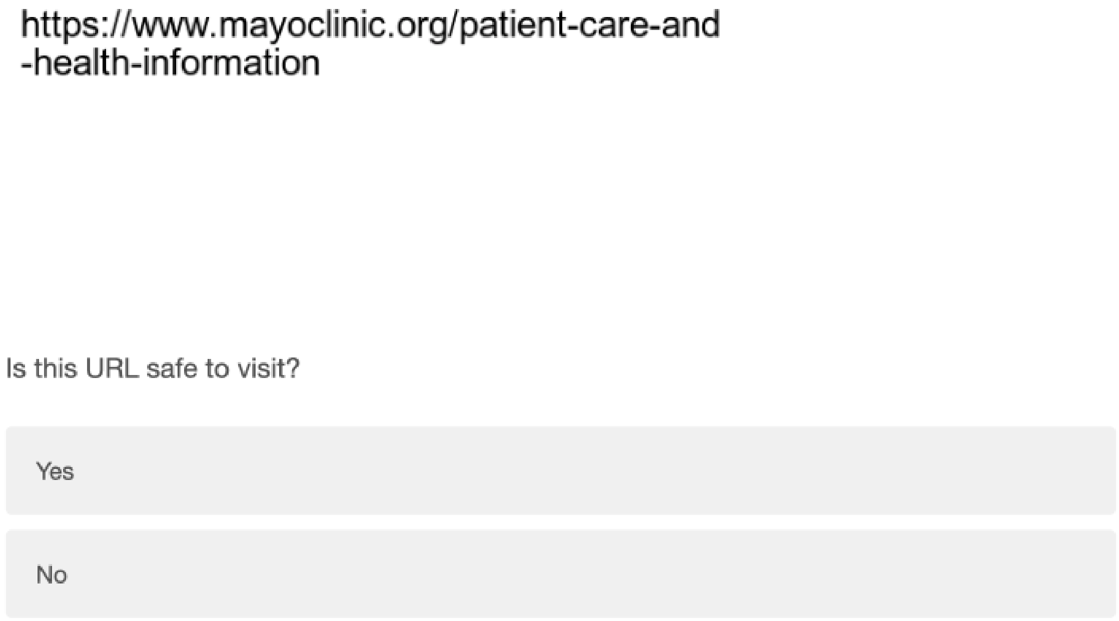
- Participants were then presented with brief instructions to set the context and warn users not to visit any URLs. These instructions read as follows:

  > "On the following screens you will be presented with a series of both safe and unsafe links. Imagine that you receive each link in an email message. For each link, indicate whether or not you believe it is safe to visit.
  >
  > **Please do not visit any URLs yourself as they may be unsafe.**"

- Next, the participants saw a series of 62 pages, each containing a URL image and the following question: "Is this URL safe to visit?" They responded by either clicking a "Yes" button or a "No" button. Figures 6.2 and 6.1 show the user interface. Qualtrics recorded both the classification for each URL, as well as the time taken to respond to each URL.

- Finally, the participant was presented with additional questions, which make up what we call the post-task questionnaire. These additional questions were designed to collect general demographic information about the participant, to assess the participant's security attitudes, to assess the participant's security knowledge, to glean insights into the participant's perceptions regarding URLs and what they deem reliable indicators of URL security, and to help evaluate experimental validity. In these preliminary results, we do not analyze responses to the post-task questionnaire.

We took precautions to improve data reliability. We ran a pilot on 20 users to gauge whether there were significant order effects. There was a prominent learning effect in the beginning so we added two additional questions to account for this. Otherwise, we did not notice order effects in the pilot. For the full experiment, we



https://www.mayoclinic.org/patient-care-and
-health-information

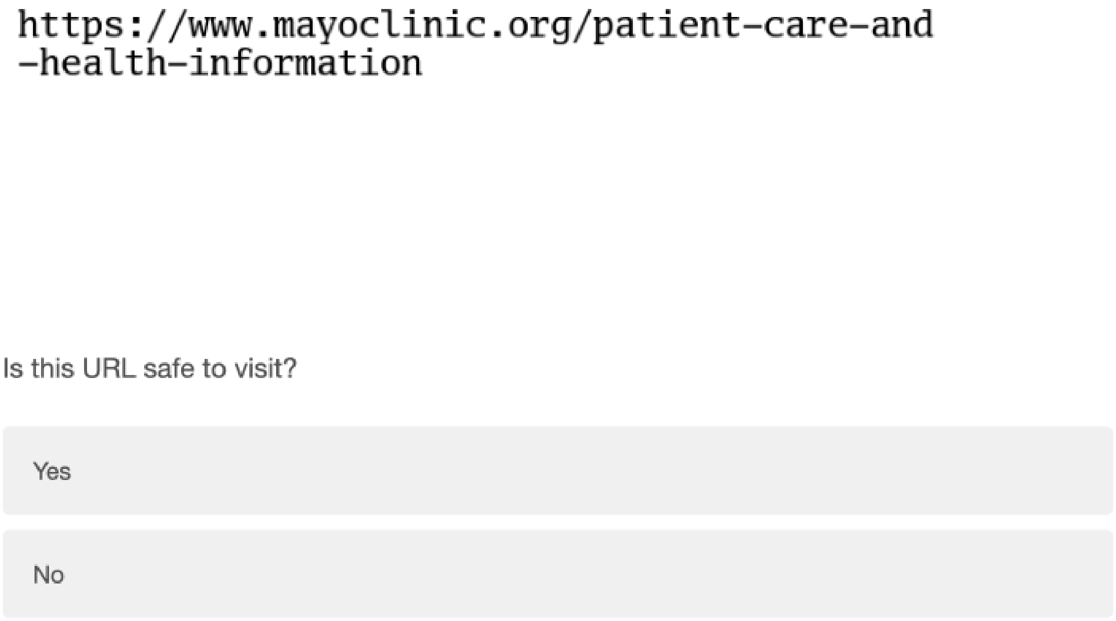
Is this URL safe to visit?

Figure 6.1: An image of the URL classification interface for the Arial font condition corresponding to URL 3 (URL class C1).

randomized the presentation order of URLs to minimize any order effects that may have taken place. While we didn't see an indication of user fatigue toward the end of the questionnaire, we simplified the wording and removed questions with free-text answers that would have been interesting to ask but were non-essential, just in case. We also asked questions to assess internal validity. Last, as noted earlier, we used *canary* URLs (C2) to detect unreliable responses.

6.7.3. Participants: Selection Criteria, Payment, Group Assignments, Outliers

We recruited participants using Amazon Mechanical Turk. We first ran a pilot on 20 participants to identify any experimental flaws; we do not use any of the data from the pilot in our results or analysis. In the full experiment we began by selecting only participants who were from the USA, who had a 99% HIT approval rate or higher, and who had completed 500 or more HITs. However, we didn't seem to be getting



https://www.mayoclinic.org/patient-care-and
-health-information

Is this URL safe to visit?

Figure 6.2: An image of the URL classification interface for the Go Mono font condition corresponding to URL 3 (URL class C1).

many responses. After the twenty-ninth participant, we changed the requirement of a 99% HIT approval rate to 96%. We obtained 240 participants over the course of two days, April 15–16, 2020. We paid all participants, but we did not use the data for 24 of the 240 participants during analysis as they classified one or more canary URLs (C2) incorrectly.

We told participants it would take approximately 10 minutes to complete the task. This time was calculated based on the results of a pilot run. We used the effective minimum wage of $11.80 [207] as a basis for setting the payment amount of $2.00; Amazon charged a $0.80 overhead for each HIT. In the interest of sharing what we've learned after we began performing this experiment, we note that, although we used Amazon Mechanical Turk for our study, there exist alternatives, including ones designed with research in mind. For example, Profilic claims they are a more ethical platform that provides better quality results. [49]

6.7.4. Groups

Participants were assigned to one of four groups. Each group was shown 62 images of URLs with equal representation of URLs from each class. Each group was shown all four URLs from class C1 (*dummy* URLs), all four URLs from class C2 (*canary* URLs), and two of the four URLs from each of the remaining 27 classes. The average length of URLs were approximately the same between the groups.

The first two groups were shown URL images that used the *regular* font (Arial). Web browsers use the sans-serif font for displaying text in address bars and status bars. We picked Arial since it has been used as a default sans-serif font on Firefox, Chrome, and Edge on Windows. [69] Mac OS X has used Helvetica as the default sans-serif font on their browsers and Ubuntu has used the font called sans-serif. While browser fonts change over time, as far as we can tell, many browsers still use these fonts or fonts that are visually similar to them. The weakness of all these fonts is that they present URLs in a fashion that makes users susceptible to ASCII homograph attacks. Namely, the lowercase version of *l* and uppercase version of *i* look extremely similar—and the *rn* character string looks similar to *m*. As many users interact with web browsers and check their email using web-based email services, it seemed appropriate to use Arial for our study.

The third group and fourth group were presented with URL images that used the *monospaced* font (Go Mono). Monospaced fonts have a fixed width. While the two fonts are dissimilar, the most notable difference is that the the character strings used in the ASCII homograph attacks we mentioned earlier are likely easier to detect with the *monospaced* font.

6.7.5. Measures and Analysis

To do statistics, we measures three things:

- The first measure, classification correctness, represents whether a URL is classified correctly by a participant; it takes a value of 1 if the participant classified the URL correctly and 0 otherwise.

- The second measure, the time spent to classify the URL, is the number of seconds that elapsed between the URL's presentation and when the user first clicked the safe or unsafe button.

- The third measure, the time spent per character to classify the URL, is simply the time spent to classify the URL divided by the number of characters in the URL.

While we did not do so in this preliminary analysis, in the full work, we will check for and remove outliers regarding temporal data. During some URL classifications, participants may have been distracted by a crying child, stepped away to take a call or get a glass of water, and so forth, translating to an extremely long classification time. While the data does not suggest such instances happened often, such instances would have had a large impact on any aggregate statistics we did if we did not account for them.

Analysis was done in R. We applied Pearson's chi-squared test with R's version of Yates' continuity correction using the chisq.test function.

Section 6.8
Results

Figures 6.3, 6.4, and 6.5 show key findings as a function of the URL class with the *regular* font and *monospaced* font differences shown with separate bars.

First, we calculated the median length of *safe* URLs, excluding *safe gatekeeper* URLs and *canary* URLs (C3–C14). We then applied Pearson's chi-squared test,

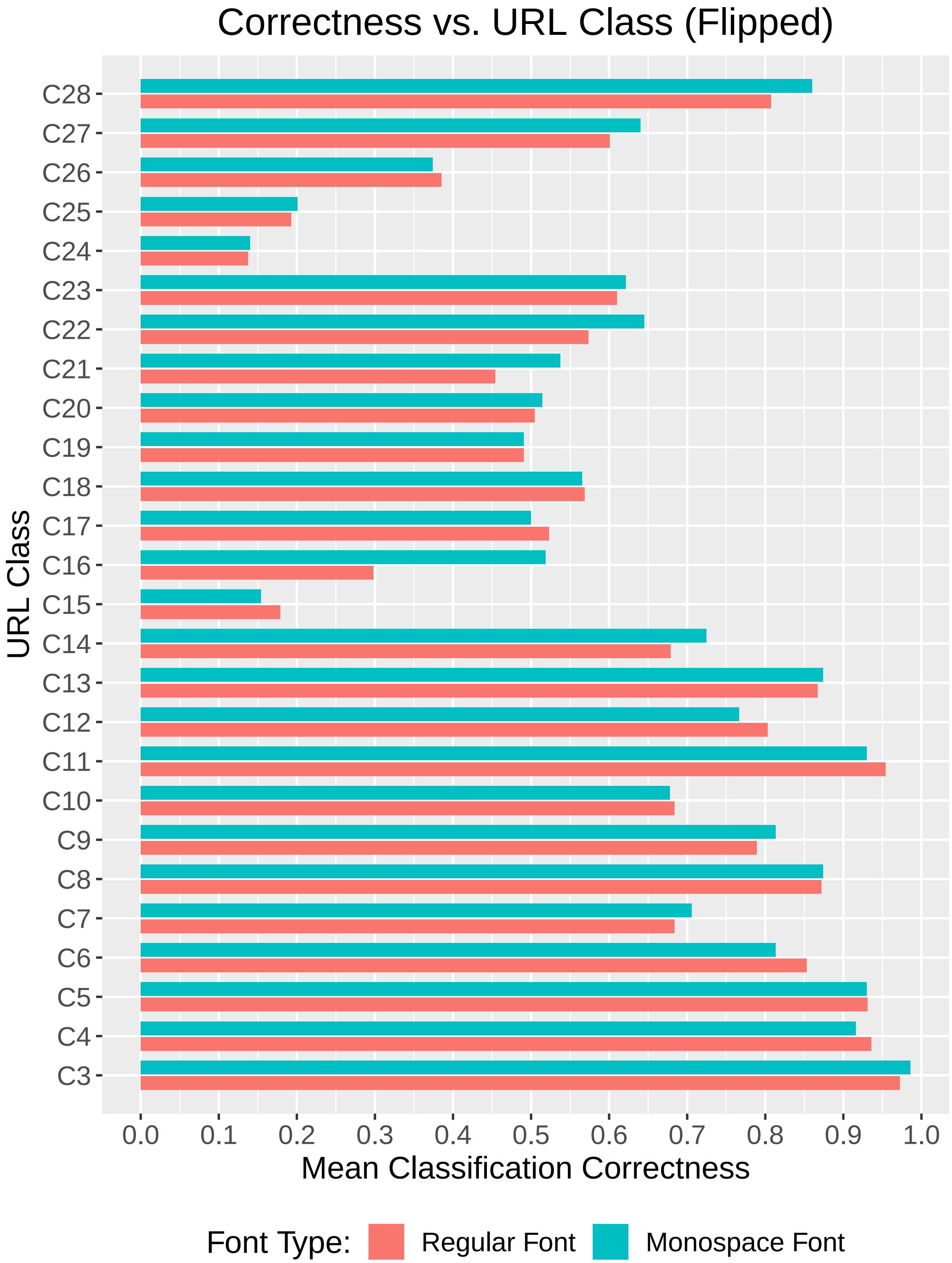


Figure 6.3: Correctness vs. class condition.

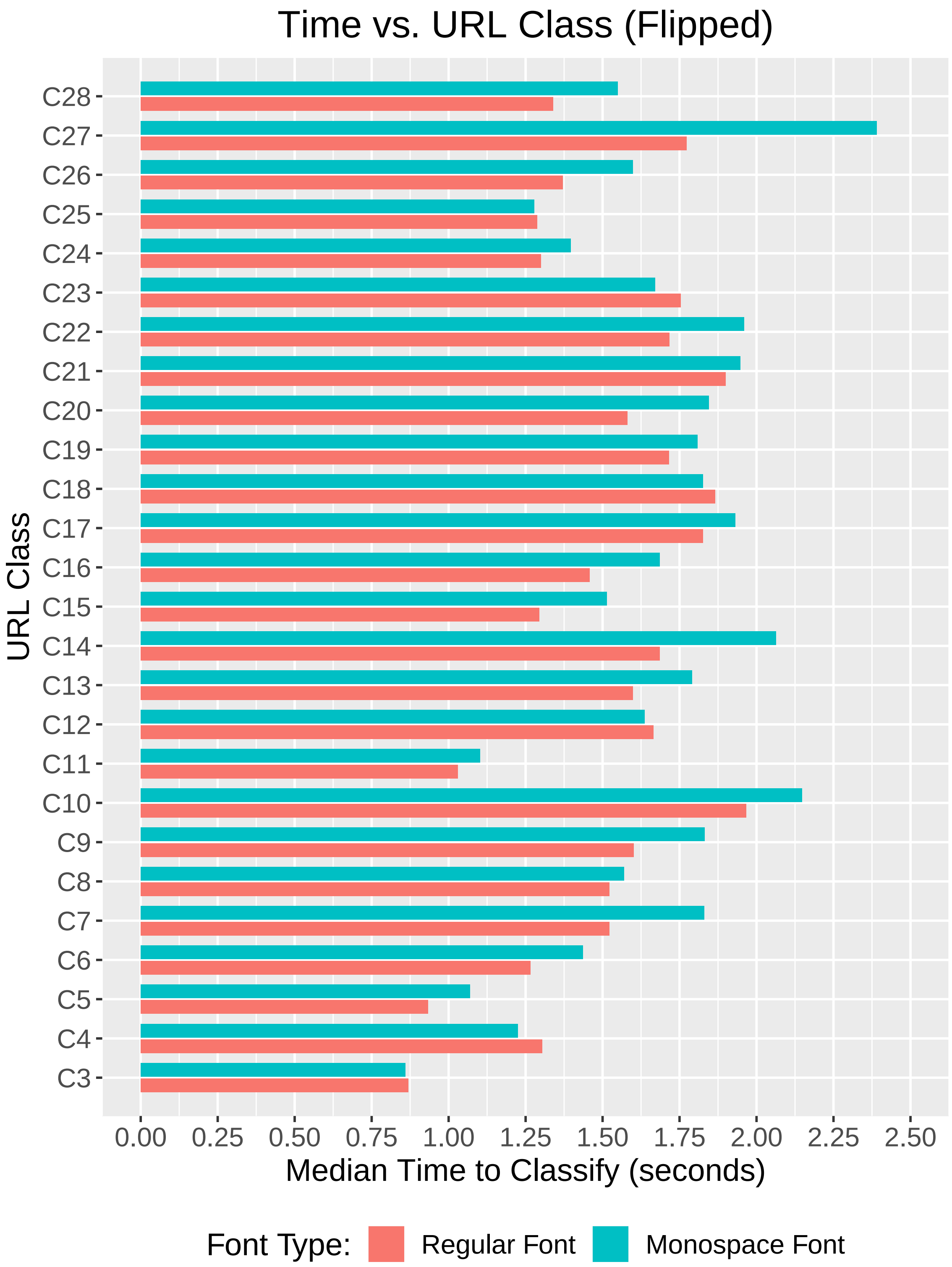


Figure 6.4: Median time taken to classify URL vs. class condition.

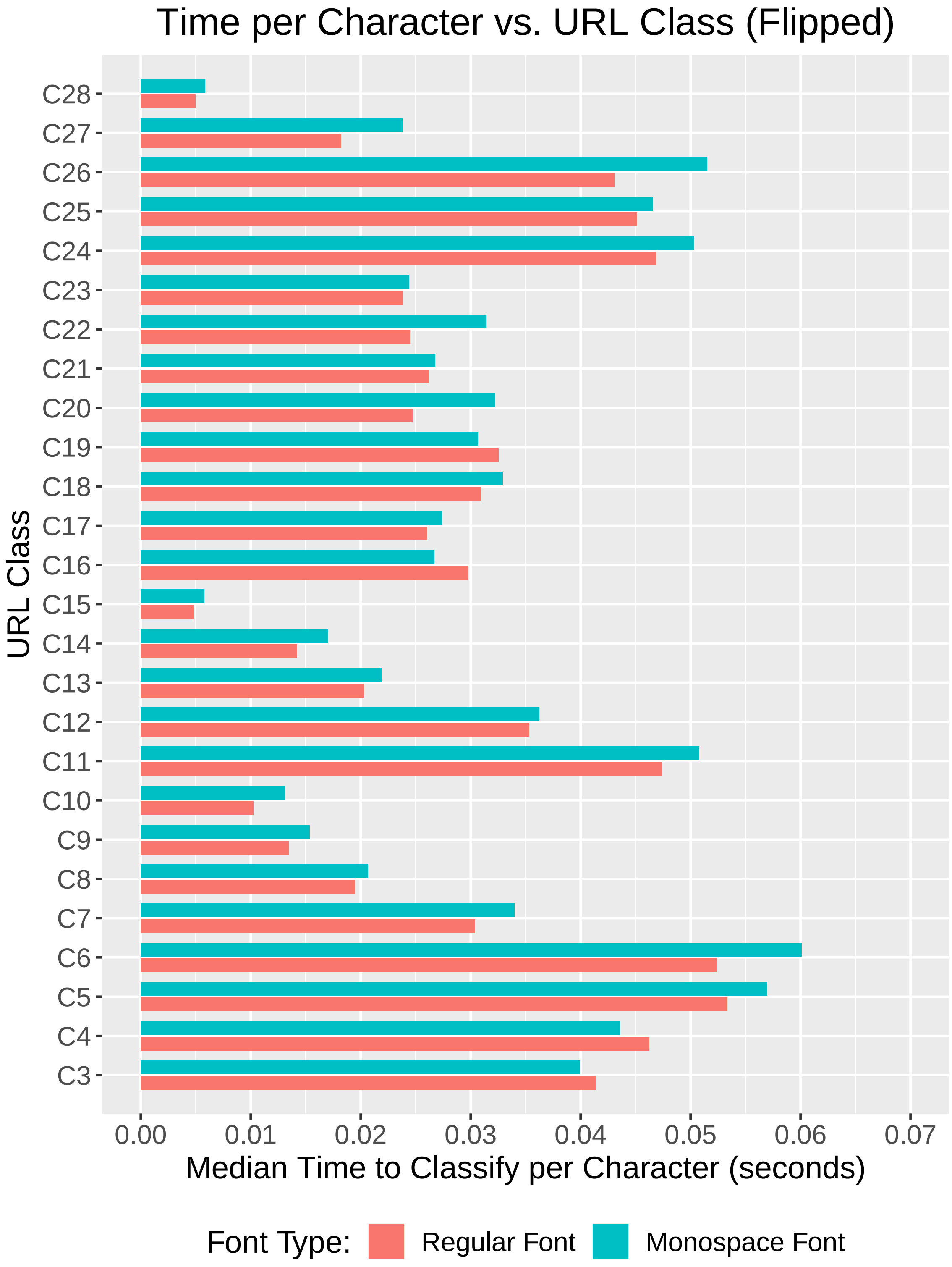


Figure 6.5: Median time taken to classify per character vs. class condition.

comparing classification correctness of URLs shorter than the median length with correctness of URLs longer than the median length. We found that there was a significant difference between the classifications ($\chi^2 = 53.13, p = 3.121 * 10^{-13}$). Using the same approach for *unsafe* URLs, again excluding *gatekeeper* URLs (C16–C27), we also saw a significant difference ($\chi^2 = 46.598, p = 38.716 * 10^{-12}$).

We compared the responses for *positive unsafe* URLs (C24) with *negative unsafe* URLs (C26). We found a statistically significant difference between the two ($\chi^2 = 33.364, p = 7.643 * 10^{-9}$).

We compared the responses for *safe gatekeeper* URLs (C15) with the responses for *unsafe gatekeeper* URLs (C26). We found no statistically significant difference between the two ($\chi^2 = 0.06065, p = 0.8055$).

We also compared responses between *monospaced* and *regular* fonts for *unsafe homograph* URLs (C16) and found that there was a statistically significant difference between the two ($\chi^2 = 20.849, 4.969 * 10^{-6}$).

Section 6.9

Analysis

Our results support hypotheses ***H1***, ***H3***, and ***H4***, but we do not have evidence to support hypothesis ***H2***, as seen in Table 6.2. The Pearson's chi-squared test showed no significant difference between the *safe gatekeeper* URLs and *unsafe gatekeeper* URLs with regard to responses. However, not all users knew of the *gatekeeper* services and this requires further inquiry. We note that the time taken to parse gatekeeper URLs was also extremely low; this may suggest that people give up on trying to classify *gatekeeper* URLs fairly quickly due to the sheer amount of complexity involved in parsing them.

From Figure 6.5, we see a difference in median time to classify URLs with a reduc-

Hypothesis	*Result*
H1: Participants are better at classifying safe short URLs than they are at classifying long URLs..	True
H2: Participants respond to safe and unsafe gatekeeper URLs differently.	False
H3: Participants classify unsafe positive-valence URLs as safe more than they do unsafe negative-valence URLs.	True
H4: Participants more accurately identify unsafe URLs in monospaced font than they do unsafe URLs in a regular font.	True

Table 6.2: Results of our hypotheses. This table explains which hypotheses we found evidence to support.

tion in time taken per character as URL length increases. This, again, may suggest that complex URLs make people give up prematurely. However, from Figure 6.4, the preliminary results do not suggest a peak in time taken to classify URLs, as we had seen with long, complex URLs in the eye-tracking study from the previous chapter. That said, this is only preliminary analysis; we have not fully removed outliers related to classification time.

We were surprised to see that participants performed worst on *short* URLs with *positive*, *neutral*, and *negative* valence (C14, C15, C16) given that these URLs do not involve some sophisticated URL obfuscation technique.

Section 6.10

Limitations

As with any study of this kind, there are a number of limitations. Below, We state these limitations and what steps we have taken to address them below.

We have only presented preliminary analysis. Most notably, the temporal results and analysis are quite rough and require further removal of outliers.

We included four dummy URLs at the beginning of our study that were not used for analysis. These URLs were just meant to acclimatize the participant to the interface and reduce learning effects. All participants saw the exact same 4 dummy URLs. So, there is the risk that exposure to those URLs may have impacted users' classifications for URLs that appeared later in the experiment, as well as the time taken to classify those URLs. Based on the pilot runs, however, we did not see significant order effects other than those associated with simply learning the interface. So, we felt the inclusion of dummy URLs is a net benefit with regard to experimental validity.

Ignoring the results of anyone who incorrectly classified a URL in class C2 may have biased our results. Some users may have legitimately felt the URLs should be classified as unsafe. Overall, we felt it was best to remove data associated with these participants as there was a higher risk of the data being unreliable if we had included them.

Any notion of URL safety will have flaws. We discuss and justify our notions of safety at length in Subsection 6.5.1.

URLs were displayed in images and many spanned multiple lines, which may have affected the way users parsed them. In practice, URLs are displayed across either one or more lines depending on the application domain. This may limit the external validity.

There may have been order effects. However, we did not observe order effects in the pilot, aside from the initial learning phase. And we randomized the order as a precautionary measure.

Research has shown the demographics of MTurk workers are not representative of the general population, e.g., [53]. For example, there is a skew toward people who make less money, who are male, and who are younger. We do not claim that the subpopulation of users we looked at is fully representative of all users, but we do feel it's a reasonable approximation.

There is the concern that workers may not have taken the task seriously. However, we took precautions by using selection criteria that required participants to have completed 500 or more tasks with a HIT rate of above 96%. We also used *canary* URLs in class C2 to detect and remove participant data that seemed unreliable. Additionally, we note that there is a strong incentive for MTurk workers to take the task seriously as their performance on each HIT is, in general, tied to their likelihood of getting work on the platform in the future.

Participants may behave differently in the real world. We tried to ensure that the context was clear and that the phrasing was easy to understand. We also asked questions to get respondents' perceptions of their performance relative to their performance in the real world. Participants generally said that they performed the experimental task as they would outside of the experimental context. That said, survey data is not always reliable, and we do expect people likely erred on the side of caution in our study. This sort of limitation is common to studies of this kind.

The phishing attacks did not perfectly resemble attacks in the wild. While we did consult phishing corpora and other studies to get an idea of the range of attacks used in practice, we created many URLs ourselves with the aim of understanding how users perceive and respond to URLs. Our focus was not on understanding exactly how susceptible users are to existing URL obfuscation URLs, but rather their susceptibility to URL obfuscation techniques in general and how various features affect susceptibility. With this aim in mind, we felt it was better to construct URLs of a certain form instead of just using URLs from, say, Phishtank [137]. This is a common approach used in other studies, e.g., [6]. We discuss this topic further in Subsection 6.5.1.

Once a user selected a response to a URL, the response was locked in and they had no opportunity to revise their response. Though the *dummy* URLs should have helped in acclimatizing the user to the interface, there will be a small fraction of incorrect classifications due to this decision. Overall, we felt that the benefits of reducing user fatigue and reducing unintended order effects justified our approach. Additionally, given that only a small fraction of users misclassified *canary* URLs for any reason, we do not believe this decision produced many misclassifications. Also, as mentioned we did not consider the data of users who misclassified *canary* URLs in our analysis.

Section 6.11 Conclusion

In this chapter, we presented our preliminary results from an ongoing study on how users assess URLs. Based on the initial results, it does not seem like participants treated *gatekeeper* that had safe URLs embedded in them as a redirect link significantly differently than *gatekeeper* URLs that had unsafe URLs embedded in them as a redirect link. Participants classified simple URLs containing a domain name that comprised two English words, possibly hyphenated, as safe much higher than we were expecting; and the valence of the words seemed to have an impact on how participants classified the URL, as we did expect. As one might expect, *monospaced* fonts had a large impact on users' ability to detect homograph attacks; however, it was not as large of a difference as we expected.

Chapter 7

A Logic for Mismorphisms

Security problems often stem from differential representations of reality wherein something that holds in one representation fails to hold in another. Our team chose the term *mismorphisms* to express these disconnects. This thesis is very much a study of mismorphisms; we have sought to learn why they occur, to study their ramifications, and to develop solutions that may help eliminate them or at least mitigate their effects. In this chapter, we focus on our recent work on mismorphisms. First, we briefly review a semiotic model we used earlier to represent mismorphisms, primarily to capture circumvention scenarios. We then motivate and discuss our more recent work on building a logical representation of mismorphisms. Finally, we demonstrate how this logical representation can be used to classify the underlying causes of a variety of real-world security issues.

Section 7.1 Introduction

Security problems often arise from one or more mismatches between what people believe about something, the representation of that thing within, say, a system or document, and the reality regarding that thing: A security practitioner may choose a password composition policy because they think it will promote the creation of strong passwords, overlooking the frustration it will cause users in practice and how that frustration may induce circumvention. A user's interpretation of security information may diverge from how a security practitioner expects the user to interpret that information. A security vulnerability in code may reflect a disconnect between implementors' and designers' assumptions. A system may assume data that is input into the system is expressed using a given type or unit of measurement, but depending on local context, that it may differ. If we dive deep enough, security problems almost always come down to one or more mismatches or, more precisely, what we call *mismorphisms*—"mappings that *fail* to preserve structure" [180].

If mismorphisms indeed lie at the heart of security issues, then understanding mismorphisms, developing a suitable model to express them, and then cataloging them may help in eliminating them or at least dealing with the problems stemming from them. In this chapter, we seek to build a simple, flexible, and usable model for expressing the underlying causes of security issues. We begin by reviewing our earlier work on mismorphisms, which utilized semiotic triads to model circumvention scenarios. We then explain our thought process for extending this work to create a logical model of mismorphisms, and we present this logical model. We then demonstrate how this logical model is capable of capturing the underlying causes of a variety of security problems, discuss directions for future work, and conclude.

Section 7.2

A Brief Background on Semiotics

Semiotics is the study of signs, processes that involve signs, and how meaning is conveyed through signs [210]. A sign may be a sound, an image, a smell, or anything else from which a sentient being extracts meaning. For a simple example, a person may see a stop sign while driving and know that means they should slow down and come to a stop. Semiotic models aim to explain these and other phenomena. Two of the most prevalent semiotic models are: the dyadic model proposed by Ferdinand de Saussure, which includes a signifier and a signified; and the triadic model proposed by Charles Sanders Peirce, which includes a sign, an object, and an interpretant. [44]. The *Stanford Encyclopedia of Philosophy* [131] provides a primer on Peirce's work.

Ogden and Richards presented the semiotic triad [132, 202] to capture the relationship between three nodes: the referent (the thing being referred to), the thought or reference (the object evoked by the referent), and the symbol (the object used to represent the thought), as seen in Figure 7.1. When a writer writes, the referent—the thing the writer is trying to express—induces a thought based on the writer's knowledge of language, who the writer thinks the reader will be and how they might interpret it, the writer's state of mind, and so forth. The thought evokes a symbol that is supposed to express the referent. Similarly, when a reader reads a word, the word or symbol evokes a thought based on the reader's general knowledge, their understanding of the context in which the word is used, and so forth. The thought is then internalized as a referent. A causal relation is established between the word (the symbol) and the thought (the reference). And a relation is also established between the thought (the reference) and the the referent. However, there is no direct relation between the symbol and the referent. Instead, there is an imputed relation established through the two sides of the triangle, not the base. Thus, we have the semiotic triad.

Before discussing our earlier work in building a semiotic triad for mismorphisms, we review some related work at the intersection of semiotics and HCI. Weir [194] discusses the need for semiotic approaches to understanding man-machine communication. Souze et al. [51] outline desired properties when designing software, advocate

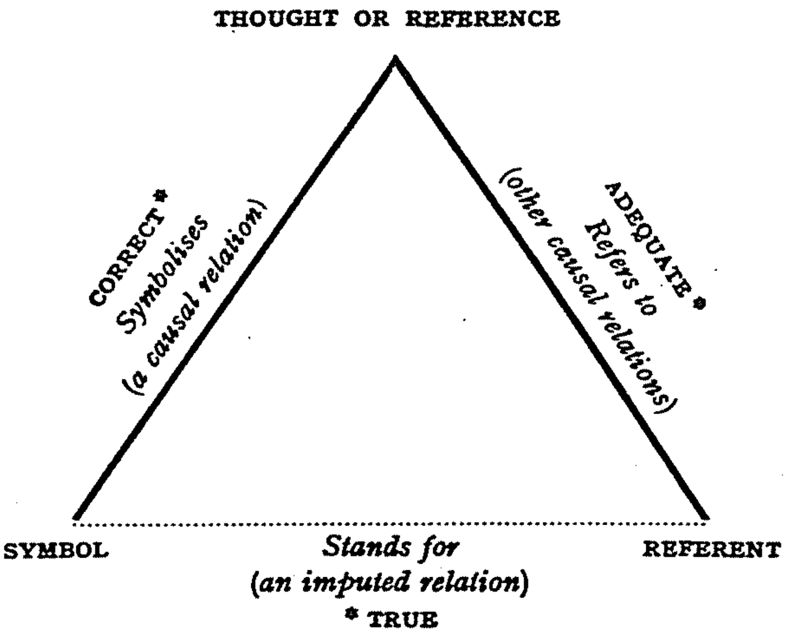


Figure 7.1: The semiotic triad. This public domain image is taken from [202] and appears on page 11 of the original 1923 publication of Ogden and Richards's *The Meaning of Meaning* [132].

for using semiotic engineering for HCI, and outline one approach. Ferreira et al. [60] look at how semiotics can be used to understand redesigns of user interfaces. They analyze three instances of redesign of a sign that is part of a user interface, and they briefly look at the contributing factors and propose that such examinations can lead to better user interface design. Andersen [12] enumerates a number of challenges semiotics-based HCI design can help address, including: "making HCI more coherent", "exploiting insights from older media," "defining the characteristic properties of the computer medium," and "situating the HCI-systems in a broader context."

Section 7.3

A Semiotic Representation of Mismorphisms

In this section, we (very) briefly review our earlier work on mismorphisms [180]. [1] The usage of mismorphism in this section is slightly different from the usage in the logical representation we later discuss. However, the essence of the two are the same; in both sections, we seek to capture how security problems arise.

As noted earlier, our semiotic model of mismorphisms is based on Ogden and Richard's semiotic triad that we covered in the previous section. This model is built with the intent of expressing circumvention scenarios. We replace the referent with a reality, the thought with a mental model, and the symbol with an IT representation as seen in Figure 7.2. In this representation,

- The reality corresponds to some truth in the real world, e.g., what actions a user may actually perform.
- The mental model corresponds to a party's beliefs regarding what the reality should be, e.g., what an admin thinks a user's permissions should be.
- The IT system representation expresses the reality as expressed in the IT system, e.g., what permissions are given to a user.

Unlike Ogden and Richards's semiotic triad, each side of the triangle now exists and links two nodes. However, this linkage is unidirectional and expresses a single mapping between representations: The reality informs the mental model. A change in the mental model may drive a party to change the IT system itself. And a change to the IT system generates a new reality. For example, a security administrator

[1] Please note that we are not treating this work [180] not as a primary thesis contribution, but as essential related work.

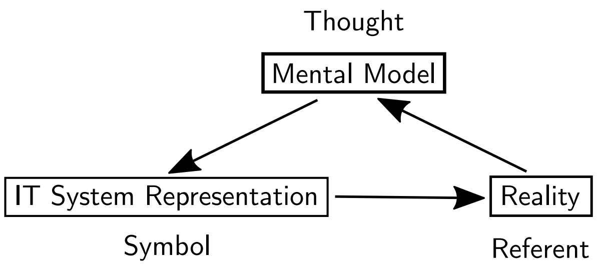


Figure 7.2: A triad for capturing circumvention scenarios.

may observe a reality in which users leave a machine unattended. This observation leads the security practitioner to the belief that there should be automatic timeouts. Thus, the security administrator may implement a policy within the IT system that automatically logs the user out if the IT system detects the user is away. And this, in turn, creates a new reality. Now, the user may become dissatisfied with this reality, e.g., because it gets in the way of delivering patient care. This reality will then drive the user to think of a way to circumvent the system. The user may then modify the IT system by, say, by attaching a mouse jiggler to the computer, which in turn generates a new reality. (In our original paper [180], there was a unidirectional relation from the mental model to the IT system representation. However, in practice, this may be bidirectional. That is, the IT system representation may inform one's mental model. This can be an important source of security vulnerabilities if security personnel rely not on the reality but the IT system representation of the reality to make future decisions.)

In semiotics, there is interest in morphisms, instances where predicates hold the same truth value across representations. However, as highlighted in the example we just discussed, what is of interest to us are instances where predicates take on different truth values across nodes of the triad. We call these *mismorphisms*. The remainder of the paper focused on exploring different classes of mismorphisms and cataloging them using the semiotic model we had developed. We found the model extremely effective in classifying circumvention scenarios.

Section 7.4

Beyond the Semiotic Triad Model of Mismorphisms

At the beginning of this chapter we posited that a notion of mismorphisms is powerful enough to capture the underlying causes of several security issues. In this section, we present the justification for a slightly different notion of mismorphisms using an alternative model that is grounded not in semiotic triads but in logic.

The semiotic model from the previous section is perfect for modeling the security circumvention scenarios we examined. However, it also somewhat constrained in representing other scenarios. Our rationale for developing an alternative model rooted in logic is as follows:

- Semiotic triads are effective in modeling scenarios where a human is interacting with a single system. The human sees the reality, thinks, and creates or modifies the system accordingly. However, things get messy when, for example, we consider multiple systems, some of which the user does not directly interact with. [180]

- Some more complicated security phenomena may require more machinery to represent. In particular, it may be useful to add a temporal dimension and to consider the effects of chaining together mismorphisms, e.g., to capture the propagation of a local security issue upward. Visually, this can become difficult to represent.

- In our semiotic representation of mismorphisms, we used mathematical logic to express the underlying predicates. Extending the notion of mismorphisms to a purely logical model while retaining the spirit of the semiotic representation seems natural.

These beliefs led us to construct the logical representation of mismorphisms presented in the next section.

Section 7.5

A Logical Representation of Mismorphisms

We now discuss our recent work on building a logical representation for mismorphisms. This representation blends temporal logic with the idea of multiple interpreters. Following this section, we demonstrate how this logical model can be used to classify underlying causes of a diverse set of security issues.

For the rest of this chapter, we will refer to a *mismorphism* as a difference in interpretation of a predicate between two or more interpreters. That is, we can think of different interpreters (e.g., a person, a system, a document, code) interpreting propositions or predicates about the world. In general, it is good when the interpretations agree and are in accordance with reality. However, when a predicate takes different truth values across different interpretations, we have a mismorphism, which may be a cause for concern.

We use the words *predicate* and *interpretation* in similar—albeit, not identical—manners to the common formal-logic meanings, e.g., as presented by Aho and Ullman [5]. However, instead of a binary logic, we use a ternary logic similar to Kleene's ternary logic [93, 62]. [2] We refer to a predicate as a function of zero or more variables whose codomain is $\{T, F, U\}$ where T is true, F is false, and U is uncertain/unknown. We refer to an *interpretation* of a predicate as an assignment of values (which may include U) to variables, which results in the predicate being interpreted as T, F, or

[2] We do not specify a specific ternary logic system for evaluating predicates in this chapter.

U. A predicate is interpreted as T if after substituting all variables for their truth values, the predicate is determined to be T; it is interpreted as F if after substituting all variables for their truth values, the predicate is determined to be F; if we are unable to determine whether the predicate is T or F, the predicate is interpreted as U.

The interpretation must be done by someone (or perhaps something) and that entity is called the *interpreter*. In our model, we have a special interpreter, the oracle O, who interprets the predicate as it is in reality (in instances where there is some ground truth). Some interpreters may not have adequate information to assign values to the variables that result in the predicate being interpreted as T or F. It is is in these instances that the predicate may be interpreted as U. We use $P|_A$ to denote the interpretation of predicate P by interpreter A.

To represent mismorphisms we need a way to express scenarios where two or or more interpreters diverge in their interpretations of a predicate. That is we must define relations on the interpreters' interpretations of a predicate. Ergo, we introduce the notion of an interpretation relation.

Predicate (Interpretation Relation) Interpreters

Each interpretation relation is a k-ary relation where $k >= 2$ denotes the number of interpreters involved—and the k-ary relation is over the interpretations of the predicate by the k interpreters. The three classes of interpretation relations we are concerned with in this chapter are: the interpretation-equivalence relations ($\boxed{\underset{\text{interp}}{=}}$), the interpretation-uncertainty relations ($\boxed{\underset{\text{interp}}{\overset{?}{=}}}$), and the interpretation-inequivalence relations ($\boxed{\underset{\text{interp}}{\overset{\times}{=}}}$).[3] The interpretation relations[4] we examine are defined as follows, where each P represents a predicate and each A_i represents an interpreter:

- $P \boxed{\underset{\text{interp}}{=}} A_1, A_2, \ldots A_k$ if and only if P, as interpreted by each A_i, has a truth value that's either T or F (never U)—and all interpretations yield the same truth value.

- $P \boxed{\underset{\text{interp}}{\overset{?}{=}}} A_1, A_2, \ldots A_k$ if and only if P takes on the value U when interpreted by at least one A_i.

- $P \boxed{\underset{\text{interp}}{\overset{\times}{=}}} A_1, A_2, \ldots A_k$ if and only if P interpreted by A_i is T and P interpreted by A_j is F for some $i \neq j$.

There are a few important observations to note. One is that the oracle O always holds the correct truth value for the predicate by definition. Another is that if we only know the $\boxed{\underset{\text{interp}}{\overset{?}{=}}}$ relation applies, we won't know which interpreter is uncertain about the predicate or even how many interpreters are uncertain unless $k = 2$ and one interpreter is the oracle. Similarly, if we only know that the $\boxed{\underset{\text{interp}}{\overset{\times}{=}}}$ relation applies, we do not know where the mismatch lies unless $k = 2$. That said, knowledge that the oracle O always holds the correct interpretation, where we are dealing with facts, combined with other information can help specify where the uncertainty or inequivalence stems from. Of course, the formalism could also be changed to allow a bit more flexibility here, but we didn't see the need. Last, the $\boxed{\underset{\text{interp}}{=}}$ relation will not be true if the $\boxed{\underset{\text{interp}}{\overset{?}{=}}}$ or the $\boxed{\underset{\text{interp}}{\overset{\times}{=}}}$ interpretations are true; however, $P \boxed{\underset{\text{interp}}{\overset{?}{=}}} A_1, \ldots A_k$ and $P \boxed{\underset{\text{interp}}{\overset{\times}{=}}} A_1, \ldots A_k$ may simulatenously be true.

[3] Note that for $k = 2$, if we confine ourselves to predicates that take on only T or F values, the relation $\underset{\text{interp.}}{=}$ is an equivalence relation in the mathematical sense, as one might expect, i.e., it obeys reflexivity, commutativity, and transitivity.

[4] Technically, they are classes of interpretation relations, but this will get tedious for me to write and you to read.

The purpose of this model is to to capture mismorphisms. Mismorphisms correspond to instances where either the interpretation-uncertainty relation or interpretation-inequivalence relation apply.

It may be valuable to consider some natural extensions to this logical formalism. In select cases, we may want to consider multiple interpreters of the same role. In these instances, we could assign subscripts to distinguish roles, e.g., D, I_1, I_2, O. Also, there are temporal aspects that may be relevant. Predicates can be functions of time and so can the interpretations. While we use the v^t-style notation to represent a variable as a function of time within a predicate, we may also consider the interpreter as a function of time, e.g., $I_4^{t_3}$ means the interpretation is done by implementor I_4 at time $t = t_3$. We do not use all of these extensions in this presentation, but if we were to create a larger catalog, they would serve useful.

Section 7.6

A Catalog of Mismorphisms

In this section, we discuss numerous examples of mismorphisms, classified by their general form. First, some remarks:

- The categories are not disjoint; some mismorphisms may be placed in two or more categories. We chose the one that seemed most appropriate.

- Some mismorphisms may be linked. For example, one mismorphism may lie at the heart of another or perhaps two mismorphisms contribute to a single security issue. This makes sense as many security issues have multiple layers of complexity. We discuss this issue more in the following section.

- We also note that there are multiple ways to do this classification. For example, another natural approach may be to choose the categories based on their application domain or security and privacy context.

- Our focus here is on applying mismorphisms to a number of new security and privacy issues in different domains.

7.6.1. Breakdown of Implication

In certain circumstances, an interpreter may believe a conditional statement that fails to hold in practice—or vice versa, they may not believe a conditional statement holds when it does hold in practice. That is, we may have something of form:

$$(X \implies Y) \boxed{\overset{\times}{\underset{\text{interp}}{=}}} A, O$$

Consider the following examples:

- A prevailing belief is that users are privacy pragmatists who willingly make an informed decision to give up their privacy in exchange for services. [56] This argument, in other words, assumes that the decision to use a service implies the user is making an informed choice. Work by Draper [56], as well as by others [74, 97, 135, 189, 190] call this view into question. Draper argues that many users feel their privacy is gone and so they resign to giving up control over their data privacy.

- Turow et al [189] found that 65% of respondents to a survey believed that the existence of a privacy policy on a site meant the site would not share their information unless they gave explicit permission.

- It is often assumed that adding a privacy option on a service will only improve users' privacy. However, the user's determination of a privacy option may, itself, leak information. For example, Lewis et al. note that both options not to share or share information correlate with other demographic information. [112].

(This point provides further justification for an opt-in approach to privacy over an opt-out one, especially in situations where most users stick with defaults.)

Alternatively, the implication operation could be correct, but X may not hold, meaning nothing can be inferred about Y. (Or perhaps, we may observe the opposite direction where both the relation and X hold in practice but not within someone's mental model.)

$$((X \implies Y) \ \boxed{\underset{\text{interp}}{=}} \ A, O) \bigwedge (X \ \boxed{\underset{\text{interp}}{\overset{\times}{=}}} \ A, O)$$

For example:

- A security practitioner may assume that any user of a service who wishes to change their privacy settings will be able to do so if they know about them—and the security practitioner may further assume the user knows about the existence of those settings. In some cases, even if the former holds, the latter does not.

7.6.2. Temporal Effects

Time may influence how predicates are evaluated. An individual may lack the foresight to identify these temporal effects.

$$(X^t = X^{t+\delta}) \ \boxed{\underset{\text{interp}}{\overset{\times}{=}}} \ A, O$$

Some examples:

- As an employee changes roles, their permissions may accumulate, whereas a security practitioner might expect the permissions to be adjusted according to the role. [175]

- Time-of-check–time-of-use (toctou) bugs [211] occur when there is a delay between when something is checked and when it is used. The delay means that operations may be performed on input that used to satisfy certain properties, but no longer does so. It reflects an oversight on the part of the developer.

- Shotgun parsing [39] involves scattering the parser code—the code responsible for vetting the input to a program—across a program, which results in code being executed before it is recognized. Vulnerabilities that exist in code that are attributable the shotgun parser anti-pattern can be classified under this class of mismorphisms.

- An analog of time-of-check–time-of-use for the privacy domain is time-of-configure–time-of-use: The user may configure their privacy settings on a social networking service once, when they begin using a service. However, over time, people may join or leave the service, leaving their privacy choices outdated. Available privacy options may also change over time.

- Gaw and Felten argue that the user may choose to reuse a password for an account—i.e., select a weak password before that account is associated with sensitive information—and, by the time that account has accrued information, "they're locked into their reused password." [65].

- A similar phenomenon may be true with privacy settings. Namely, the user may choose privacy settings before sensitive information is tied to their account. By the time sensitive information is tied to their account, the user may no longer think about privacy. Moreover, in instances where the user does contemplate reconfiguring privacy settings, there's a possibility that the data in question may be perceived to already be lost and, therefore, not worth protecting.

- On the other hand, some users may have already invested significant time or effort in selecting a piece of software, downloading it, and installing it before they configure their privacy settings, compelling them to continue using the service even if it does not meet their privacy needs. That is, they may fall victim to the sunk cost fallacy [15]. Had they known of the invasive privacy settings beforehand, they may have chosen to go with a competitor.

7.6.3. A Knowledge Gap

In certain circumstances, an actor's lack of knowledge about how to interpret information may contribute to a security issue. Here, P may be a statement about, say, a system, and the interpreter may be ill-equipped to evaluate the truth of that statement, resulting in an unknown truth value under their interpretation:

$$P \boxed{\underset{\text{interp}}{\overset{?}{=}}} A, S, O$$

- Users may lack the requisite information to make informed decisions, often because that information is simply not available. It is not always clear how services safeguard user data, nor the intricacies of how that data is used in practice. Privacy policies exist, but may be exorbitantly time-consuming to read and difficult to digest [130, 115]. Moreover, they are often vague and usually subject to change. A pessimist might argue that in practice many existing interfaces and privacy policies ensure users remain uninformed while presenting the veneer of informed consent, thereby persuading their users and other actors that user data is in good hands. Another concern is that primary services or third-party services may violate privacy policies, terms of service, or users' privacy expectations. This may even be compounded by a delay in reporting violations. Collectively, these and other factors support the argument that most users do not— and, at least in the current privacy landscape cannot— have a concrete understanding of how their data is used.

- A user may lack the capability to come to a determination regarding the safety of a URL (e.g., shortened URLs, gatekeeper URLs), the legitimacy of an email, or the meaning of certificate information. While some users may seek information that informs their mental model, others may fall back on insecure behavior because it's less effort and potentially a lower perceived cost than alternatives. Even if a user seeks out information, it is possible that they may consult a resource that provides inaccurate information.

7.6.4. Projections

An interpreter A's interpretation of how interpreter B would interpret a predicate P may differ from how B actually interprets it. That is, we may have:

$$P|_B \; \boxed{\underset{\text{interp}}{\overset{\times}{\equiv}}} \; A, O$$

We recognize that there is a slight abuse of notation here. To resolve this, we can simply substitute $P|_B$ with "B's interpretation of P," to avoid the double-meaning of $P|_B$—or we could create a wrapper. Recall the oracle is always correct and so their interpretation of $P|_B$ align with what $P|_B$ actually is. In any case, here is an example of such a mismorphism:

- Actual and perceived time and effort to configure privacy settings may influence whether the user begins configuring them and whether they finish. For example, the user may be dissuaded from using an interface that appears illogical, complex, or hard to navigate. Or, as we mentioned earlier, they may simply lack the requisite knowledge to make meaningful decisions that align with their intentions. The security practitioner or others may perceive users' effort to configure their privacy settings to be minimal or ignore them all together and view the option of configuration as a binary choice.

Section 7.7

Peeling the Layers of Mismorphisms

In the previous section, we presented a (simplified) catalog of mismorphisms that are responsible for a variety of security problems. However, the power of mismorphisms as an explanatory model comes from the ability to both break down a mismorphism and study its ramifications. Identifying these causal relations allows us deconstruct and learn from existing security problems.

For example, why might a user wrongly classify an unsafe URL as safe? Well, one reason may be that their mental model of where the URL goes is flawed [6]. This can be captured as a mismorphism between security properties of the URL in the system representation and the security properties of the URL within the user's mental representation. But why does that mismorphism exist? It may be a purely visual problem, due to a poor choice of font, which can be expressed as a mismorphism between the user's mental representation and the information shown in the real-world and/or a mismorphism between the system representation and the information shown in the real-world, depending on where the problem lies. Or perhaps the user correctly interprets what characters are on the screen, but fails to extract the correct security information from those characters; this again can be represented as a mismorphism between the URL specification (or, more precisely, the layers of systems involved in resolving the URL and delivering content to the user) and the user's mental representation. But we could again ask: why is there a mismorphism between a user's mental model of URL structure and the way users are resolved in practice? And so on.

Ultimately, it is this process of recursive deconstruction of mismorphisms that reveals why a security problem truly exists. Understanding mismorphisms and the links between them is essential in addressing many of the security problems faced today.

One way to logically represent this is to consider mismorphisms as an expression and define relations and operators on mismorphisms, notably the incorporation of causal relations.

The addition of causal relations bring us one step closer to being able to represent the semiotic-triad-based model, though it contains no explicit actions. Approaches such as supplementing the notion of mismorphisms discussed here with, say, events, may provide this additional flexibility. But this is beyond the scope of this chapter.

Section 7.8
Conclusion

Recognizing the mismorphisms that produce the intent-outcome mismatches is critical in addressing those mismatches. In this chapter, we pursued a logical model of mismorphisms to complement our earlier work on representing mismorphisms using semiotic triads. We reviewed the earlier semiotic triad representation, provided rationale for developing a new model, introduced our logical representation, cataloged a variety of mismorphisms, and discussed how security problems could be represented as a chain of mismorphisms.

Chapter 8

Conclusion

In this concluding chapter, we very briefly review the aims of this thesis, discuss our contributions, and highlight key themes and takeaways that emerged over the course of our work. Last, we give our final thoughts.

Section 8.1

Chapter Contributions

The purpose of this thesis was to better understand why intent-outcome mismatches exist, to model them, and to develop solutions that help address them. The thesis covered six primary chapters, each corresponding to a project that makes a bit of headway toward this grand objective:

Chapter 2 Contributions We examined Amazon reviews for password logbooks; these reviews illustrate how well-intentioned password policies can prompt user circumvention and potentially undermine security objectives. They also shed light on the struggles regular users have in managing passwords, the perceptions and misperceptions they hold, and the failures of some existing security solutions. This work demonstrates the sheer amount of user insights we can get by scraping freely available data supplied by and about users. There's also a strong argument to be made that such reviews are voluntary and provided with little or no provocation, leading to more genuine responses than, say, a questionnaire may produce; on the flip side, they are less structured and the subpopulation of people reviewers may be skewed. While we did not adopt sophisticated automated techniques to process the data, doing so would further improve this approach. The data gathered through such studies can be extremely valuable in informing security decisions.

Chapter 3 Contributions We developed an agent-based password simulation to model password behavior and help choose a password composition policy within an organization. In response to password composition policies over different services, agents manage their passwords by memorizing them. They also write them down and reuse them to cope with the burden of complying with the policies of the many services with which they interact. These sorts of simulations serve as a pathway for decision-makers to reason about security policies and make sound decisions that achieve their intentions. They reveal how certain security decisions affect user and organizational objectives and provide a way to compare security solutions. They may even highlight instances where a system is unusable or indicate user inclination to circumvent even when specific circumventive techniques are unknown.

Chapter 4 Contributions We introduced the notion of human-computability boundaries to complement existing work in the field of language-theoretic security (LangSec). LangSec delivers a process for security design and development of parsers and protocols. However, the security of this process still very much depends on humans. We discussed ways that people fail in practice, posited that incorporating human-computability boundaries into the LangSec methodology could provide a more complete solution, and suggested threads for future work. This exploration lays the foundation for future research threads that can help system designers and implementors build systems that achieve the security properties they desire.

Chapter 5 Contributions We conducted a study where participants were tasked with classifying a series of URLs while wearing an eye tracker. The eye-tracking data we collected tells us about how users visually process URLs. Additionally, as task-invoked pupillary response reflects an increased cognitive load, the data also tell us something about users' cognitive processes. Our findings could be used to improve security awareness training, to inform how organizations choose URLs so as to improve users' security perceptions of them, and to improve security defenses. Our findings essentially tell us how to bridge the gap between what URLs users think is safe and what actually is safe.

Chapter 6 Contributions To complement our eye-tracking study, we also conducted a large-scale study over MTurk to get a better handle on various factors that influence URL processing, specifically location, language, and font. We also examined which flavors of URL attacks are most effective. Similarly to the eye-tracking study, this study provides valuable insights into how users perceive URLs, which can be used to improve security.

Chapter 7 Contributions Finally, we used a logical representation to express mismorphisms. This representation serves as a natural way to express intent-outcome mismatches, which are the focus of this thesis. Mismorphisms often manifest as security problems, e.g., as undesirable user behavior or vulnerabilities in code. We demonstrated the value of our logical representation in expressing the underlying causes of a variety of security problems.

Section 8.2

Recurring Themes

While each chapter provided its own contributions, there are also a number of important recurring themes that emerged in our collective pursuit that warrant discussion. And so, we discuss them here.

8.2.1. Unusable Security Decisions Often Induce User Circumvention

Time and time again, we observe that unusable security solutions are ineffective. Sure, unusable and stringent security solutions may "work" in the short-term, but that is usually at the cost of driving up user frustration, getting in the way of user workflow, impeding progress toward other organizational objectives, and reducing user tolerance for complying with future demands. Often, in addition to the aforementioned problems that unusable security solutions create, the solution is actually no solution at all; users become so frustrated that they circumvent and defeat the security policy or mechanism altogether. In some circumstances, the circumvention not only negates any expected security gains but it actually reduces aggregate security because the previously deployed solution was at least somewhat effective. Or perhaps the circumvention is so good that it fools security practitioners into thinking all is well, preventing them from implementing more usable solutions, ones that may have initially had lower expected security gains but would have resulted in higher actual gains.

8.2.2. Unusable Security Solutions May Create New Security Problems

In some cases, we not only find that security solutions worsen aggregate security but that the circumvention strategies that users develop create wholly new security problems. It's important to recognize that such problems do not just spontaneously appear. And they are not usually due to evil users. [4] They are the consequences of unusable security policies or mechanisms, ineffective communication between security personnel and users, and myopia. Various techniques, including those presented throughout this thesis, can help address these problems.

8.2.3. Implicit Training

Users' mental models and security behaviors are informed by their previous interactions with services. Services have significant control over how these interactions play out. Services determine the password policies users must satisfy, which affects the actual and perceived cognitive burden associated with remembering passwords and also affects user notions of what constitutes a safe password. Services supply users with the URLs that inform users' mental models regarding what constitutes a safe URL. Services send emails that inform users' mental models of what constitutes a safe email. Services that heavily use Javascript or otherwise require browser security settings to be lowered for proper functionality mean users have to either jump through hoops to stay safe on the service or opt for the most lax security settings.

That is, services—in choosing what password policies to apply, what URLs to serve, which email addresses to send emails from, whether to use attachments in emails, which security settings they require users to disable, whether to require users to share information to access things, and so forth—are implicitly training users. These kinds of interactions shape user notions of what information is acceptable to share, what security precautions should be heeded, when to ignore security warnings, who to trust, and so forth.

For the most part, security research is divorced from this implicit training. Security problems are often attributed to user ignorance, frustration, laziness, or misbehavior—or perhaps the practice of the single service on which the problem arose. However, much of users' beliefs and behaviors are inculcated through routine interactions with the many services they use. An interesting, under-explored research question that has emerged over the course of this work is: how can we identify and improve this implicit training?

8.2.4. The Security Dependency

As mentioned in our previous work on mismorphisms [180], the security practices of one service can have a notable impact on the aggregate security of another organization. That is, we mustn't only take an organization-centric view of security. It is critical to acknowledge and account for both the implicit and overt training users have received from other services and the user data available to other services or elsewhere. A commonly used phrase by security researchers and practitioners alike is "users are the weakest link." But a commonly overlooked question is: "what led them to become the weakest link?" This thesis attempted to unravel some of the interdependencies between services with regard to security. However, more must be done.

8.2.5. Collective Action

Some of the worst security problems faced today cannot be handled by a single organization. This is because, as we discussed, the security posture of an organization is not based solely on its own decisions but also those of other organizations. While the decisions of powerful entities may, in certain circumstances improve aggregate security—e.g., Google "strongly advocating that sites adopt HTTPS encryption" [168] [1]—this type of approach is not always feasible. Many problems created by collective decisions call for a collective response. One such example of a response is Let's Encrypt [72], a free-certificate authority, which came out of a collaboration between Mozilla, Cisco,

[1] Of course, there are still some services that use http, e.g., see `https://whynohttps.com`. [81]

EFF, and others, whose emergence catalyzed wider adoption of HTTPS. [2]

We believe collective action by organizations has a role more broadly. And it can be especially important for improving the implicit training that users receive. For example, what if we standardized the format of URLs among leading services? Or what if we standardized certain aspects of the URL, e.g., the query field associated with URL redirection? Doing so could reduce the variability in the URLs served to users, improve implicit training, and ultimately lead users to have a more accurate representation of how a URL is structured and where legitimate services place redirection URLs. When a URL doesn't fit into this model, the user would be better equipped to reject it without fear of false negatives, in contrast to their current inability driven by faulty models constructed from interactions with services that use a hodgepodge of URL structures.

8.2.6. Beyond a Single Objective

We mustn't think about security objectives in isolation. A security decision may waste user time, result in user error, or cause other issues. It is important to identify the trade-offs and to make an informed and holistic decision that considers the decision outcomes for all objectives, not just security. That is, if we are serious about achieving a given security objective, we must think about that objective in relation to other objectives—or perhaps as a single component of a grand objective. Even if a security solution achieves a security objective in the short term, it may be rolled back once there is a broader understanding of its ramifications. Thus, even if one is solely considered about security outcomes, the broader objectives of the organization must be taken into account to optimize decision-making.

There are also a number of often-overlooked costs and considerations of security

[2] We note that various concerns have been expressed over Let's Encrypt, e.g., Scott Helme explains a common criticism of Let's Encrpyt, and then provides a defense. [77]

decisions. Does the security decision erode the trust that users place in security personnel? How will the decision affect users' mental models? How do we evaluate whether the security objective is actually working—and are we using the right measure to do this evaluation? And is the measure reliable? Simply thinking about these sorts of questions early in the design process can help in achieving the intended outcomes.

Section 8.3 Future Work

We discuss research threads that could be spun off the work presented in this thesis.

In Chapter 2, we collected and analyzed Amazon reviews of password logbooks. We believe reviews are an underutilized resource for usable security information; they provide a large, freely accessible data set comprising records of users' security beliefs, security behaviors, and security woes. It would be interesting to examine other products and their reviews. In particular, we think it would be enlightening to analyze reviews of mouse jigglers, USB sticks and contraptions used to emulate mouse movements to defeat auto-logout systems. Additionally, there exist plenty of other services besides Amazon from which we can learn about users. Users share their thoughts on blogs, news aggregator sites, forums, and various other platforms. Harnessing such data responsibly can reveal, at scale, information that is often inaccessible or hard to get via other means.

In Chapter 3, we discussed our work on building an agent-based simulation to simulate users' password behaviors and assist security practitioners in choosing and assessing password policies. Agent-based modeling can be an extremely effective tool for modeling the ramifications of security decisions if it accurately captures users' cognitive processes, behaviors, and defects—e.g., short attention spans, heuristics, and finite memories. Moreover, the growing availability of security data only broadens the scope of what can be studied. With regard to our password simulation, it would be interesting to examine and model more modern struggles that users face with managing passwords by incorporating two-factor authentication and password managers into the simulation. Password managers are usually considered among security practitioners to be an ideal solution. However, as we discovered in our study on password logbooks, users have concerns, with varying legitimacy, about password managers; they are concerned about the safety of password managers, whether they can share passwords with relatives, and whether processes exist to pass on passwords to relatives after death. Additionally, there is the question of how users should navigate the various options available and select a password manager. Another topic that may be interesting to pursue is modeling the effects of implicit training, which we discussed in the previous section. For one example, it may be useful to use simulation to study the impact that organizational decisions have on how users assess the safety of emails and URLs, especially given the growing availability of data on these topics.

The ideas presented in Chapter 4 lay the foundation for future work on how to discover and account for human-computability boundaries. This area of research certainly requires more exploration. There has been much progress in building approaches to secure code but they often rely on humans behaving according to some ideal; in practice, human behavior diverges from this ideal. Identifying precisely what properties humans must have to achieve the envisioned security objectives, identifying what properties they *do* have, and developing ways to achieve those properties in practice are natural next steps.

Chapters 5 and 6 deal with two studies on how users classify URLs. The first study involved having participants classify URLs in a lab while wearing an eye tracker; in the second study, we had workers classify URLs as we recorded their responses and time taken to respond. Both studies could benefit from exploration of additional conditions associated with inducing different moods, introspection, a sense of urgency, and cognitive load. Both studies presented the user with URLs images wherein the URL may span more than one line; in many applications, however, users see URLs on just one line. It would be interesting to examine what impact this makes and how only being able to partially view a URL affects safety evaluations. The eye-tracking study used fairly coarse areas of interest; smaller boxes, e.g., one for each domain label in the fully qualified domain name, could help us better understand how users parse URLs. We plan to continue the Amazon Mechanical Turk study to explore the impact that country of residence (USA vs. India) and language (English vs. Hindi) have on how URLs are classified. However, we also see other directions for future work. We explored valence in the MTurk study, but other factors, like arousal as used in the VAD model [121] may have also had an impact on URL classification. Our Amazon Mechanical Turk study was fairly broad in scope; it would now be nice to focus in on a few areas to explore the more subtle phenomena that cannot be detected with our sample size. Last, we used a fixed corpus of 116 URLs in the MTurk study. It would be interesting to conduct a new study where each participant is presented with its own, entirely unique set of URLs. This would provide more generalizable results and likely allow for the study of more phenomena. For example, we could more easily disentangle URL simplicitly from shortness. Such an approach could also assist in providing valuable data to feed into a machine learning algorithm to simulate how users process URLs, which in turn could be used within an agent-based simulation to assess the impacts of implicit training.

Chapter 7 discusses our work on building a logical model for mismorphisms and creating a catalog of mismorphisms. This catalog can be expanded. Additionally, it may benefit from more machinery to express actions associated with mismorphisms. Ultimately, we would like to see a collaborative tool built atop this model for identification and classification of mismorphisms. We believe this would be extremely useful in understanding and addressing security problems in practice.

Section 8.4

Final Thoughts

In this thesis, we pursued the problem of bridging the gap between intent and outcome. Each chapter made progress toward this end goal. Many chapters also showcased a broadly applicable technique using a specific problem scenario. In Chapter 2, we demonstrated the value of applying grounded theory to a large corpus of freely available data to gather insights into users' beliefs, behaviors, and struggles; we used this technique to learn why users resort to password logbooks as a password management device, to identify user misperceptions, and to learn what security practitioners might be missing with regard to usable security and password management. In Chapter 3, we demonstrated the utility of using agent-based simulation to assess security solutions and identify circumvention behaviors by exploring the question of how best to set password policies. In Chapter 4, we argued that approaches used in language-theoretic security could be strengthened by also considering human-computability boundaries. In Chapter 5, we demonstrated how eye tracking can deliver valuable ground-truth data about how users visually process security information by looking at how they parse URLs. In Chapter 6, we did a complementary study, informed by our eye-tracking study, to examine other phenomena associated with how users parse URLs, the factors at play, and URL features that affect processing. Both these approaches have limitations, but by applying them together, many of the limitations can be at least partially addressed. Last, in Chapter 7, we explored a logical model to express mismorphisms based on our earlier semiotic work on mismorphisms; we believe this logical representation for mismorphisms—perhaps with some refinements—can serve as a unifying model to express the underlying causes of a vast array of security problems.

The contribution of each chapter bring us one step closer to ensuring that the security and privacy intentions of individuals—security practitioners, coders, users, etc.—and also collectives, such as organizations, are realized in practice. However, as communicated in the previous section, there is a lot of interesting work left to be done, including: harnessing the continually growing, freely accessible data sets containing valuable, untapped security information; applying agent-based simulations to new scenarios with new data and insights; identifying and accounting for human-computability boundaries; using eye tracking to study URLs at a finer level of granularity and also to explore other use cases; utilizing the reach of Amazon Mechanical Turk to explore conditions that are hard to explore via other means (e.g., impact of first language on URL classification); and building on our logical representation of mismorphisms. We are particularly interested in seeing the mismorphism model extended to a usable, collaborative tool to identify and catalog mismorphisms. We believe the most effective way to tackle security issues is to address the mismorphisms that contribute to them—and this is precisely what such a tool could achieve.

In closing, we believe we have delivered vital contributions that help to bridge the gap between decision-maker intent and the outcomes of decisions driven by said intent. While we in no way claim to have fully solved this problem, the work presented in this thesis does lay a path forward. By deconstructing the overarching problem of addressing the intent-outcome mismatch into smaller, more tractable subproblems, we provided concrete findings, demonstrated broadly applicable techniques, and produced tools and models for security-minded decision-making. Of course, given the sheer scope of the challenge, there is more to do. However, we believe this thesis provides a foundation upon which future researchers can build, just as previous researchers laid a foundation for us.

Appendix A

URL Corpus Used in Chapter 6

We list the URLs used in our URL corpus. Each class contains 4 URLs. For each class, we list the class name and the features associated with the class, preceded by a "#" symbol, followed by the 4 URLs that belong to the class (with the URL ids we used included). The features groups, in order, correspond to the safety of the URLs, the length of the URLs, the top-level domain of the URLs, whether the URL contains a path, and other features.

```
# Form of classes is as follows:
# class: (un)safe, length, (non-)com, (non-)path, other


# C1: mixed, mixed, mixed, mixed, dummy
1 https://www.theverge.com/2020/4/3/21206400/apple-tax-amazon
   -tv-prime-30-percent-developers
2 https://www.nylon-llama.com
3 https://www.mayoclinic.org/patient-care-and-health-
   information
4 http://www.nationalcupcakeday.ca/spayneuter.ontariospca.ca/
   cgi-bin/GodSo/GodSo/googledriveeesss/nD/
```

```
# C2: safe, short, com, non-path, www, canary
5 https://www.youtube.com
6 https://www.google.com
7 https://www.amazon.com
8 https://www.facebook.com

# C3: safe, short, com, non-path, www,
9 https://www.instagram.com
10 https://www.kayak.com
11 https://www.cvs.com
12 https://www.sprint.com

# C4: safe, short, com, www, path, www
13 https://www.hulu.com/welcome
14 https://www.nike.com/air-max
15 https://www.vox.com/recode
16 https://www.starbucks.com/gift

# C5: safe, short, com, non-path, non-www, 2-level domain
17 https://twitter.com
18 https://slack.com
19 https://time.com
20 https://gizmodo.com

# C6: safe, short, com, non-path, non-www, 3-level domain
```

```
21 https://postcalc.usps.com
22 https://about.ikea.com
23 https://stats.nba.com
24 https://shop.nordstrom.com

# C7: safe, medium, com, path
25 https://www.etsy.com/c/art-and-collectibles?ref=catnav-66
26 https://www.delta.com/flight-status/schedule/STL/AUS
   /2020-03-24
27 https://www8.hp.com/us/en/home.html
28 https://apnews.com/6202bebc5b0f5fa80904d06e50b8429f

# C8: safe, long, com, path
29 https://www.yelp.com/search?cflt=restaurants&find_loc=San
   %20Francisco%2C%20CA
30 https://www.politico.com/news/2020/03/24/congress-
   coronavirus-emergency-package-146066
31 https://www.t-mobile.com/cell-phone/samsung-galaxy-s20-5g?
   sku=610214663405
32 https://www.dominos.com/en/pages/order/#!/locations/search
   /?type=Delivery

# C9: safe, very long, com, path
33 https://www.espn.com/nfl/story/_/id/28871296/2020-nfl-free
   -agency-trade-grades-bill-barnwell-tracks-every-big-
   signing-move
```

```
34 https://www.bloomberg.com/news/articles/2020-03-24/youtube
   -to-limit-video-quality-around-the-world-for-a-month?srnd=
   premium
35 https://us.norton.com/products/norton-360-antivirus-plus?
   inid=nortoncom_nav_norton-360-antivirus-plus_homepage:home
36 https://www.bedbathandbeyond.com/store/product/brita-reg-
   soho-5-cup-water-filtration-pitcher/3328522?categoryId
   =12119

# C10: safe, extremely long, com, path
37 https://www.washingtonpost.com/politics/irs-to-begin-
   issuing-1200-coronavirus-payments-april-9-but-some-
   americans-wont-receive-checks-until-september-agency-plan-
   says/2020/04/02/8e0cfc84-751e-11ea-85cb-8670579b863d_story
   .html?tid=pm_pop&itid=pm_pop
38 https://www.americanexpress.com/us/customer-service/
   digital/amex-mobile-app.html?intlink=us-en-hp-hero-cta-all
   -AmexAppJan2020-16032020
39 https://www.hiltonhonors.com/en_US/20200106_2011/landing/?
   cid=OM,MB,MO2011_53ee0.18fff.ffffffff055dd14.32795ff4_All
   ,MULTIPR,Interact,Multipage,SingleLink
40 https://www.wired.com/story/coronavirus-interview-larry-
   brilliant-smallpox-epidemiologist/#intcid=
   recommendations_default-popular_ede7315b-73cb-4c2d-a56e-4
   be6572b5f68_popular4-1
```

```
# C11: safe, short, non-com, non-path
41 https://www.nih.gov
42 https://www.irs.gov
43 https://www.harvard.edu
44 https://www.coursera.org

# C12: safe, medium, non-com, path
45 https://nh.craigslist.org/d/garage-moving-sales/search/gms
46 https://www.npr.org/programs/fresh-air/
47 https://www2.ed.gov/fund/grants-college.html?src=image
48 https://www.cornell.edu/about/mission.cfm

# C13: safe, long, non-com, path
49 https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing
   /symptoms.html
50 https://www.khanacademy.org/science/physics/forces-newtons
   -laws#newtons-laws-of-motion
51 https://www.pbs.org/video/amazon-empire-the-rise-and-reign
   -of-jeff-bezos-xpco5j/
52 https://news.stanford.edu/2020/03/18/climate-change-means-
   extreme-weather-predicted/

# C14: safe, very long, non-com, path
53 https://en.wikipedia.org/wiki/Martin_Luther_King_Jr.#
   Selma_voting_rights_movement_and_”Bloody_Sunday”,_1965
```

```
54 https://www.who.int/news-room/detail/03-03-2020-shortage-
   of-personal-protective-equipment-endangering-health-
   workers-worldwide
55 https://addons.mozilla.org/en-US/firefox/?utm_source=www.
   mozilla.org&utm_medium=referral&utm_campaign=nav&
   utm_content=firefox
56 https://www.nasa.gov/press-release/nasa-spacex-invite-
   media-to-first-crew-launch-to-station-from-america-since
   -2011

# C15: safe, extremely long, com, www, path, gatekeeper
57 https://urldefense.proofpoint.com/v2/url?u=https-3A__www.
   google.com&d=vER2ua&c=oUjqpuXqgtpWNC19zQwiz&r=
   dVLkjRi4dYkcqmlbnyrohGZggmTpF7O39Sy8BrXDmVj&m=
   pGF2UTOoa117QkpGdSxzgS5uEsPKexbsigi3ECUXMCE&s=
   qB2nveLVjUr47p6PFvDC4wu9Z8XTtjHuyV9vKU2gbEd&e=
58 https://urldefense.proofpoint.com/v2/url?u=https-3A__www.
   facebook.com&d=jpUXTH&c=1WXqUzkUyKcNxYPlM4a3M&r=
   pJsfsAs5wrIfp3HK5j4oWOH9NRsWHUpPy3SE1HkI2NO&m=3
   uVpEFKx8zWoLnfdngzBbpnU5lsVagAgb2iOtUU38Qy&s=
   bWskFlXWAYWUO3L4ntktaMMCYkLZqwFq5mWKMO52sz8&e=
59 https://nam01.safelinks.protection.outlook.com/?url=https
   %3A%2F%2Fwww.amazon.com&data=02%7C01%7Cnettie.danielson
   .83411%40gmail.com%7Cba4509dc2eac227bae927c1be39a5bf1%7
   C2efe2af7675e9f555df1eef400bf5e69%7C0%7C0%7
   C905766577259738989&sdata=1
   bU9GNY5ArebQJeTc3DyZ023ftmd6mbmYfu36X9znDI%3D&reserved=0
60 https://nam01.safelinks.protection.outlook.com/?url=https
   %3A%2F%2Fwww.youtube.com&data=02%7C01%7Cdorothy.carducci
   .63288%40gmail.com%7C7666a6108d9d53a002baa1c825b9972d%7
   Cb9b1424d7082cb1c503d114b00fdc77d%7C0%7C0%7
   C891102336761180193&sdata=
   GvYNkcXzZzhdkZwRzEF7YAbA5GSefTaLs6l1Ey0Ro06%3D&reserved=0

# C16: unsafe, short - very long, com, mixed, ASCII homograph
    (l->I, m->rn)
61 https://www.zilIow.com
62 https://www.hornedepot.com/c/diy_projects_and_ideas
63 https://www.yeIp.com/search?find_desc=Delivery&find_loc=
   Philadelphia%2C%20PA
64 https://www.arnericanexpress.com/us/credit-cards/category/
   travel-rewards/?inav=menu_cards_pc_travelrewardscards

# C17: unsafe, short - very long, com, mixed, combosquatting
   (https://x1.x2.x3...xk.com is legitimate; substitute xk
   for xk'=r(xk')s)
65 https://www.adobe-update.com
66 https://www.appleid.reset-apple-id.com/password/verify/
   appleid
67 https://www.wellsfargo-accounts.com/checking/compare-
   checking-accounts/?linkLoc=fn
```

```
68 https://www.nytimes-global.com/interactive/2020/us/
   coronavirus-us-cases.html?action=click&module=Spotlight&
   pgtype=Homepage

# C18: unsafe, short - very long, com, mixed, infix domain
   label (e.g., https://ebay.errorpayments.com)
69 https://www.youtube.yt-red.com
70 https://netflix.new-customer-promos.com/one-month-free/
71 https://www.paypal.smart-help.com/us/smarthelp/article/can
   -i-cancel-a-paypal-payment-faq637
72 https://pages.ebay.sc-help-pages.com/az/en-us/seller-
   center/service-and-payments/managed-payments-on-ebay.html#
   new-payment-exp

# C19: unsafe, short - very long, com, mixed, wrong top-level
    domain
73 https://www.spotify.vg/
74 https://www.walmart.tk/m/deals/home-savings/home
75 https://www.bankofamerica.name/credit-cards/products/cash-
   back-credit-card/
76 https://www.tripadvisor.io/Attraction_Review-g35805-
   d2485153-Reviews-Adagio_Teas-Chicago_Illinois.html

# C20: unsafe, short - very long, com, mixed, domain-in-
   domain (google.com.evil.com)
77 https://www.att.com.att-wl.com
```

```
78 https://www.dropbox.com.dropbox-basic.com/login
79 https://www.linkedin.com.linkedin-join.com/?trk=
   guest_homepage-basic_nav-header-logo
80 https://www.airbnb.com.homes-sydney-australia.com/s/Paris/
   homes?refinement_paths%5B%5D=%2Fhomes&search_type=
   section_navigation

# C21: unsafe, medium - very long, com, mixed, domain-in-
   domain with hex obfuscation (google.com.evil.com)
81 https://www.att.com%2Eatt-wl%2E%63%6F%6D
82 https://www.dropbox.com%2Edropbox-basic%2E%63%6F%6D/login
83 https://www.linkedin.com%2Elinkedin-join%2E%63%6F%6D/?trk=
   guest_homepage-basic_nav-header-logo
84 https://www.airbnb.com%2Ehomes-sydney-australia%2E%63%6F%6
   D/s/Paris/homes?refinement_paths%5B%5D=%2Fhomes&
   search_type=section_navigation

# C22: unsafe, short - very long, com, mixed, user@host
85 https://www.imdb.com@imdb-go.com
86 https://www.microsoft.com@en.us-microsoft-365.com?rtc=1
87 https://medium.com@kiarajoshi12.delhi-baking.com/bakery-
   courses-in-delhi-9b4fe65e484c
88 https://www.theguardian.com@world.id0518492538.com/2020/
   mar/31/how-will-the-world-emerge-from-the-coronavirus-
   crisis
```

```
# C23: unsafe, medium - very long, com, mixed, user@host with hex obfuscation
89 https://www.imdb.com@imdb-go%2E%63%6F%6D
90 https://www.microsoft.com@en%2Eus-microsoft-365%2E%63%6F%6D?rtc=1
91 https://medium.com@kiarajoshi12%2Edelhi-baking%2E%63%6F%6D/bakery-courses-in-delhi-9b4fe65e484c
92 https://www.theguardian.com@world%2Eid0518492538%2E%63%6F%6D/2020/mar/31/how-will-the-world-emerge-from-the-coronavirus-crisis

# C24: unsafe, short, com, non-path, positive valence
93 https://www.farm-living.com
94 https://www.carnivalpark.com
95 https://www.sweetest-pets.com
96 https://www.joybakery.com

# C25: unsafe, short, com, non-path, neutral valence
97 https://www.datageek.com
98 https://www.saber-footwork.com
99 https://www.zippytransit.com
100 https://www.tomahawk-gear.com

# C26: unsafe, short, com, non-path, negative valence
101 https://www.furydemolition.com
102 https://www.inferno-garbage.com
```

```
103 https://www.pain-insecticide.com
104 https://www.predator-torpedo.com

# C27: unsafe, long and very long, com, path, URL redirection
105 https://t-info.mail.adobe.com/r/?id=hc43f43t4a,afd67070,
    affc7349&p1=https://bit.ly/6qfGDB0
106 https://business.facebook.com/ads/creativehub/select/?
    redirect_uri=https%3A%2F%2Fbit.ly/p5wv65V
107 https://facebook.com+login_oage&amp;welcome_to_facebook=
    true&amp;timestamp=42837643@bit.ly/g7Sxl9F
108 https://www.google.com/url?sa=D&q=https://appengine.
    google.com/_ah/logout%3Fcontinue%3Dhttps://bit.ly/6qfGDB0

# C28: unsafe, extremely long, com, path, SafeLinks/
    URLDefense
109 https://urldefense.proofpoint.com/v2/url?u=https-3A__www.
    woodlever.com&d=HuzkNr&c=Wyn5MtLBFsCvfq5tvVkMj&r=
    ULuIzfFDHCA6fQaDW9cy81gw3bwm2OXQOwjTkjxyAbA&m=
    fAHL7lahBYN92daxL5ryLQkfIFhVsTzbiBWG4OoaAW4&s=
    ledvDZtYPN2jv7Lljwj98TC810o5o2grsfbFdKY0lSs&e=
110 https://urldefense.proofpoint.com/v2/url?u=https-3A__www.
    corner-2Dstaircase.com&d=YcWbNS&c=dfqwHb6rUSxr9QLDuZskC&r=
    n01RUBHqXPlcKGwjnL1bu80P2i954p3fsSedSjPxufA&m=
    GoSStpqKZmfaFWsPNsJwknYTNuSjFuTkmzIYwh5PYaJ&s=
    tFYv1nIObUfi6BW9RAwEyRVNFm4Wk6aCPXWh7Ryf7cg&e=
111 https://nam01.safelinks.protection.outlook.com/?url=https
    %3A%2F%2Fwww.bullet-torque.com&data=02%7C01%7Caida.wisener
    .29460%40gmail.com%7Ca52a2dea9aefd5b559dd4ca86af47ae6%7
    C0102c3231815fe6e6457b782e2f2ef09%7C0%7C0%7
    C443255411014997303&sdata=
    nfqslHeNlVhwcsvJrpfC9autuaccdyc9i8oJjt3l8kN%3D&reserved=0
112 https://nam01.safelinks.protection.outlook.com/?url=https
    %3A%2F%2www.lancerarmor.com&data=02%7C01%7Cyael.luke
    .55301%40gmail.com%7C10fb992491b663c172d0fd5a5a23abf3%7
    C87bad0ac2c0140c885522b13564ed149%7C0%7C0%7
    C650765915103583656&sdata=0
    VW3kcrYxnZuWD64GkPw21DVxhdaOYRLxFX653VRifS%3D&reserved=0

# C29: unknown safety, short, mixed, path, shortened URLs
113 https://t.co/n7oywunJ9K
114 https://goo.gl/bl3ixr
115 https://bit.ly/0B3GQ1g
116 https://tinyurl.com/4XkHbrw
```

Bibliography

[1] About Amazon Verified Purchase Reviews. Amazon. https://www.amazon.com/gp/help/customer/display.html?nodeId=201145140. Accessed: 03-11-2017.

[2] LANGSEC: Language-theoretic Security: "The View from the Tower of Babel". http://langsec.org.

[3] Business ID leak via Creative Hub redirect. September 2019. https://philippeharewood.com/business-id-leak-via-creative-hub-redirect/.

[4] Anne Adams and Martina Angela Sasse. Users are Not the Enemy. *Communications of the ACM*, 42(12):40–46, 1999.

[5] Al Aho and Jeff Ullman. Foundations of Computer Science: C Edition, Chapter 14. http://infolab.stanford.edu/~ullman/focs.html, July 1994.

[6] Sara Albakry, Kami Vaniea, and Maria K. Wolters. What is This URL's Destination? Empirical Evaluation of Users' URL Reading. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376168. URL https://doi.org/10.1145/3313831.3376168.

[7] Nora Alkaldi and Karen Renaud. Why Do People Adopt, or Reject, Smartphone Password Managers? In *1st European Workshop on Usable Security, Darmstadt*, volume 18, pages 1–14, 2016.

[8] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, 82:69 – 82, 2015. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2015.05.005. URL `http://www.sciencedirect.com/science/article/pii/S1071581915000993`.

[9] Kholoud Althobaiti, Kami Vaniea, and Serena Zheng. Faheem: Explaining URLs to people using a Slack bot. In *Symposium on Digital Behaviour Intervention for Cyber Security (AISB)*.

[10] Prashant Anantharaman, J. Peter Brady, Ira Ray Jenkins, Vijay H. Kothari, Michael C. Millian, Kartik Palani, Kirti V. Rathore, Jason Reeves, Rebecca Shapiro, Syed Tanveer, Sergey Bratus, and Sean W. Smith. Intent as a Secure Design Primitive. In C.A. Kamhoua, L.L. Njilla, A. Kott, and S.S. Shetty, editors, *Modeling and Design of Secure Internet of Things (To Appear)*, chapter 24. Wiley, 2020. ISBN 9781119593362. URL `https://books.google.com/books?id=fljywAEACAAJ`.

[11] Prashant Anantharaman, Vijay Kothari, J. Peter Brady, Ira Ray Jenkins, Sameed Ali, Michael C. Millian, Ross Koppel, Jim Blythe, Sergey Bratus, and Sean W. Smith. Mismorphism: The Heart of the Weird Machine. In *Security Protocols Workshop XXVII (To Appear)*. Springer International Publishing, 2020.

[12] Peter Bøgh Andersen. What semiotics can and cannot do for HCI. *Knowledge-Based Systems*, 14(8):419–424, 2001.

[13] John R Anderson. Problem Solving and Learning. *American Psychologist*, 48 (1):35, 1993.

[14] Nalin Asanka Gamagedara Arachchilage, Steve Love, and Konstantin Beznosov. Phishing threat avoidance behaviour: An empirical investigation. *Computers in Human Behavior*, 60:185–197, 2016.

[15] Hal R Arkes and Catherine Blumer. The psychology of sunk cost. *Organizational behavior and human decision processes*, 35(1):124–140, 1985.

[16] Dirk Balfanz, Glenn Durfee, Diana K Smetters, and Rebecca E Grinter. In search of usable security: Five lessons from the field. *IEEE Security & Privacy*, 2(5):19–24, 2004.

[17] Barracuda. Barracuda phishline. `https://www.barracuda.com/products/phishline`, 2019.

[18] David Basin, Saa Radomirovic, and Lara Schmid. Modeling Human Errors in Security Protocols. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 325–340. IEEE, 2016.

[19] Jackson Beatty. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological bulletin*, 91(2):276, 1982.

[20] Adam Beautement, M. Angela Sasse, and Mike Wonham. The Compliance Budget: Managing Security Behaviour in Organisations. In *Proceedings of the 2008 New Security Paradigms Workshop*, NSPW '08, pages 47–58, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-341-9. doi: 10.1145/1595676.1595684. URL `http://doi.acm.org/10.1145/1595676.1595684`.

[21] Giampaolo Bella and Lizzie Coles-Kemp. Layered Analysis of Security Ceremonies. In *IFIP International Information Security Conference*, pages 273–286. Springer, 2012.

[22] Z. Benenson, A. Girard, N. Hintz, and A. Luder. Susceptibility to URL-based Internet attacks: Facebook vs. email. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 604–609. IEEE, March 2014. doi: 10.1109/PerComW.2014.6815275.

[23] Zinaida Benenson, Freya Gassmann, and Robert Landwirth. Unpacking Spear Phishing Susceptibility. In Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson, editors, *Financial Cryptography and Data Security*, pages 610–627, Cham, 2017. Springer International Publishing. ISBN 978-3-319-70278-0.

[24] Jennifer Romano Bergstrom and Andrew Schall. *Eye Tracking in User Experience Design*. Elsevier, 2014.

[25] Tim Berners-Lee, Larry Masinter, Mark McCahill, et al. Uniform Resource Locators (URL). RFC 1738, RFC Editor, December 1994. URL `https://tools.ietf.org/html/rfc1738`.

[26] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform Resource Identifiers (URI): Generic Syntax. RFC 3986, RFC Editor, August 1998. URL `https://tools.ietf.org/html/rfc3986`.

[27] David Beymer and Daniel M. Russell. WebGazeAnalyzer: A System for Capturing and Analyzing Web Reading Behavior Using Eye Gaze. In *CHI '05 Extended*

Abstracts on Human Factors in Computing Systems, CHI EA '05, pages 1913–1916, New York, NY, USA, 2005. ACM. ISBN 1-59593-002-7. doi: 10.1145/1056808.1057055. URL `http://doi.acm.org/10.1145/1056808.1057055`.

[28] David Beymer, Daniel Russell, and Peter Orton. An Eye Tracking Study of How Font Size and Type Influence Online Reading. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 2*, BCS-HCI '08, pages 15–18, Swindon, UK, 2008. BCS Learning & Development Ltd. ISBN 978-1-906124-06-9. URL `http://dl.acm.org/citation.cfm?id=1531826.1531831`.

[29] Manuel Blum and Santosh Vempala. The Complexity of Human Computation: A Concrete Model with an Application to Passwords. *arXiv preprint arXiv:1707.01204*, 2017.

[30] Jim Blythe. A dual-process cognitive model for testing resilient control systems. In *Resilient Control Systems (ISRCS), 2012 5th International Symposium on*, pages 8–12. IEEE, 2012.

[31] Jim Blythe and L Jean Camp. Implementing mental models. In *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, pages 86–90. IEEE, 2012.

[32] Jim Blythe, Ross Koppel, and Sean W Smith. Circumvention of Security: Good Users Do Bad Things. *IEEE Security & Privacy*, 11(5):80–83, 2013.

[33] Jim Blythe, Vijay Kothari, Sean Smith, and Ross Koppel. Usable Security vs. Workflow Realities: Work in Progress. In *Workshop on Usable Security (USEC 2018)*, 2018.

[34] Catherine M Bohn-Gettler and David N Rapp. Depending on My Mood: Mood-Driven Influences on Text Comprehension. *Journal of Educational Psychology*, 103(3):562, 2011.

[35] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank 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.

[36] Margaret M. Bradley, Laura Miccoli, Miguel A. Escrig, and Peter J. Lang. The pupil as a measure of emotional arousal and autonomic activation. *Psychophysiology*, 45(4):602–607, 2008. doi: 10.1111/j.1469-8986.2008.00654.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.2008.00654.x`.

[37] Michael Bratman et al. *Intention, plans, and practical reason*, volume 10. Harvard University Press Cambridge, MA, 1987.

[38] Sergey Bratus, Michael Locasto, Meredith Patterson, Len Sassaman, and Anna Shubina. Exploit Programming: From Buffer Overflows to "Weird Machines" and Theory of Computation. {*USENIX; login:*}, 36(6):13–21, December 2011.

[39] Sergey Bratus, Meredith L Patterson, and Dan Hirsch. From "shotgun parsers" to more secure stacks. *Shmoocon*, 2013.

[40] Daniel Bruneau, M. Angela Sasse, and John Mccarthy. The Eyes Never Lie: The Use of Eye Tracking Data in HCI Research. In *In Proceedings of the CHI'02 Workshop on Physiological Computing*. ACM Press, 2002.

[41] William E. Burr, Donna F. Dodson, and W. Timothy Polk. Special Publication 800-63: Electronic Authentication Guideline, 2004.

[42] Christine Burton and Meredyth Daneman. Compensating for a Limited Working Memory Capacity During Reading: Evidence from Eye Movements. *Reading Psychology*, 28(2):163–186, 2007. doi: 10.1080/02702710601186407. URL `https://doi.org/10.1080/02702710601186407`.

[43] Gamze Canova, Melanie Volkamer, Clemens Bergmann, and Benjamin Reinheimer. NoPhish App Evaluation : Lab and Retention Study. In *NDSS Workshop on Usable Security*. Internet Society, 2015.

[44] Daniel Chandler. Semiotics for Beginners. `http://visual-memory.co.uk/daniel/Documents/S4B/sem02.html`.

[45] Yee-Yin Choong. A cognitive-behavioral framework of user password management lifecycle. In *Human Aspects of Information Security, Privacy, and Trust*, pages 127–137. Springer, 2014.

[46] Rosetta Code. Levenshtein distance. `http://rosettacode.org/wiki/Levenshtein_distance`.

[47] B. Jack Copeland. The Church-Turing Thesis. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2019 edition, 2019.

[48] Bennett Cyphers, Alexei Miagkov, and Andrés Arrieta. Privacy Badger Now Fights More Sneaky Google Tracking. `https://www.eff.org/deeplinks/2018/10/privacy-badger-now-fights-more-sneaky-google-tracking`, October 2018.

[49] Ekaterina (Katia) Damer. Stop using MTurk for research. *Prolific*, July 2019. `https://blog.prolific.co/stop-using-mturk-for-research/`.

[50] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The tangled web of password reuse. In *Symposium on Network and Distributed System Security (NDSS)*, 2014.

[51] Clarisse Sieckenius de Souza, Simone Diniz Junqueira Barbosa, and Raquel Oliveira Prates. A semiotic engineering approach to user interface design. *Knowledge-Based Systems*, 14(8):461–465, 2001.

[52] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why Phishing Works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 581–590, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124861. URL `http://doi.acm.org/10.1145/1124772.1124861`.

[53] Djellel Difallah, Elena Filatova, and Panos Ipeirotis. Demographics and dynamics of mechanical turk workers. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 135–143, 2018.

[54] Julie S. Downs, Mandy Holbrook, and Lorrie Faith Cranor. Behavioral Response to Phishing Risk. In *Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit*, eCrime '07, pages 37–44, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-939-5. doi: 10.1145/1299015.1299019. URL `http://doi.acm.org/10.1145/1299015.1299019`.

[55] Christine E. Drake, Jonathan J. Oliver, and Eugene J. Koontz. Anatomy of a Phishing Email. In *CEAS 2004 - First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA*, 2004.

[56] Nora A Draper. From Privacy Pragmatist to Privacy Resigned: Challenging

Narratives of Rational Choice in Digital Privacy Debates. *Policy & Internet*, 9 (2):232–251, 2017.

[57] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. You'Ve Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1065–1074, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357219. URL http://doi.acm.org/10.1145/1357054.1357219.

[58] Carl Ellison. Ceremony Design and Analysis. Cryptology ePrint Archive, Report 2007/399, 2007. https://eprint.iacr.org/2007/399.

[59] Michael Fagan and Mohammad Maifi Hasan Khan. Why Do They Do What They Do?: A Study of What Motivates Users to (Not) Follow Computer Security Advice. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 59–75. USENIX Association, 2016.

[60] Jennifer Ferreira, Pippin Barr, and James Noble. The semiotics of user interface redesign. In *Proceedings of the Sixth Australasian conference on User interface-Volume 40*, pages 47–53. Australian Computer Society, Inc., 2005.

[61] Ian Fette, Norman Sadeh, and Anthony Tomasic. Learning to Detect Phishing Emails. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 649–656, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242660. URL http://doi.acm.org/10.1145/1242572.1242660.

[62] Melvin Fitting. Kleene's Three Valued Logics and Their Children. *Fundam.*

Inf., 20(1-3):113–131, March 1994. ISSN 0169-2968. URL http://dl.acm.org/citation.cfm?id=183529.183533.

[63] Dinei Florêncio, Cormac Herley, and Paul C Van Oorschot. Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts. In *Proc. USENIX Security*, 2014.

[64] Joseph P Forgas. Mood effects on decision making strategies. *Australian Journal of Psychology*, 41(2):197–214, 1989.

[65] Shirley Gaw and Edward W Felten. Password Management Strategies for Online Accounts. In *Proceedings of the Second Symposium on Usable Privacy and Security*, pages 44–55. ACM, 2006.

[66] Thomas Gilovich, Dale Griffin, and Daniel Kahneman. *Heuristics and Biases: The Psychology of Intuitive Judgment*. Cambridge University Press, 2002.

[67] Sanjay Goel, Kevin Williams, and Ersin Dincelli. Got Phished? Internet Security and Human Vulnerability. *Journal of the Association for Information Systems*, 18(1):2, 2017.

[68] Joseph H. Goldberg, Mark J. Stimson, Marion Lewenstein, Neil Scott, and Anna M. Wichansky. Eye Tracking in Web Search Tasks: Design Implications. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02, pages 51–58, New York, NY, USA, 2002. ACM. ISBN 1-58113-467-3. doi: 10.1145/507072.507082. URL http://doi.acm.org/10.1145/507072.507082.

[69] Scott Granneman. Web Browser Font Defaults. https://www.granneman.com/webdev/coding/css/fonts-and-formatting/web-browser-font-defaults.

[70] Paul A. Grassi, James L. Fenton, Elaine M. Newton, Ray A. Perlner, Andrew R. Regenscheid, William E. Burr, Justin P. Richer, Naomi B. Lefkovitz, Jamie M. Danker, Yee-yin Choong, Kristen K. Greene, and Mary F. Theofanos. NIST Special Publication 800-63B: Digital Identity Guidelines: Authentication and Lifecycle Management, 2017.

[71] Paul A. Grassi, Michael E. Garcia, and James L. Fenton. NIST Special Publication 800-63-3: Digital Identity Guidelines, 2017.

[72] Internet Security Research Group. Let's Encrypt. `https://letsencrypt.org`.

[73] Elizabeth Ha and David Wagner. Do Android Users Write about Electric Sheep? Examining Consumer Reviews in Google Play. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 149–157. IEEE, 2013.

[74] Eszter Hargittai et al. Facebook privacy settings: Who cares? *First Monday*, 15(8), 2010.

[75] Eiji Hayashi and Jason Hong. A Diary Study of Password Usage in Daily Life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2627–2630. ACM, 2011.

[76] Rosa Heckle. Security Dilemma: Healthcare Clinicians at Work. *IEEE Security & Privacy*, 9(6):14–19, 2011.

[77] Scott Helme. Let's Encrypt are enabling the bad guys, and why they should. `https://scotthelme.co.uk/lets-encrypt-are-enabling-the-bad-guys-and-why-they-should/`, 3 2017.

[78] Cormac Herley. So Long, and No Thanks for the Externalities: The Rational

Rejection of Security Advice by Users. In *Proceedings of the 2009 New Security Paradigms Workshop*, pages 133–144. ACM, 2009.

[79] Kyung Wha Hong, Christopher M. Kelley, Rucha Tembe, Emerson Murphy-Hill, and Christopher B. Mayhorn. Keeping up with the joneses: Assessing phishing susceptibility in an email task. volume 57, pages 1012–1016, 2013. doi: 10.1177/1541931213571226. URL https://doi.org/10.1177/1541931213571226.

[80] Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International conference on Knowledge Discovery and Data Mining*, pages 168–177. ACM, 2004.

[81] Troy Hunt. Why No HTTPS? https://whynohttps.com.

[82] Alexa Huth, Michael Orlando, and Linda Pesante. Password Security, Protection, and Management. *United States Computer Emergency Readiness Team*, 2012.

[83] Jukka Hyönä, Robert F Lorch Jr, and Johanna K Kaakinen. Individual Differences in Reading to Summarize Expository Text: Evidence From Eye Fixation Patterns. *Journal of Educational Psychology*, 94(1):44, 2002. doi: 10.1037//0022-0663.94.1.44. URL https://www.doi.org/10.1037//0022-0663.94.1.44.

[84] Jukka Hyönä, Robert F Lorch Jr, and Mike Rinck. Chapter 16 - Eye Movement Measures to Study Global Text Processing. In Jukka Hyönä, Ralph Radach, and Heiner Deubel, editors, *The Mind's Eye*, pages 313–334. North-Holland, Amsterdam, 2003. ISBN 978-0-444-51020-4. doi: 10.1016/

B978-044451020-4/50018-9. URL http://www.sciencedirect.com/science/article/pii/B9780444510204500189.

[85] Philip G Inglesant and M Angela Sasse. The True Cost of Unusable Password Policies: Password Use in the Wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 383–392. ACM, 2010.

[86] David E Irwin and Gregory J Zelinsky. Eye movements and scene perception: Memory for things observed. *Perception & Psychophysics*, 64(6):882–895, 2002.

[87] Blake Ives, Kenneth R Walsh, and Helmut Schneider. The domino effect of password reuse. *Communications of the ACM*, 47(4):75–78, 2004.

[88] Christian Johansen, Tore Pedersen, and Audun Jøsang. Towards Behavioural Computer Science. In *IFIP International Conference on Trust Management*, pages 154–163. Springer, 2016.

[89] PN Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Harvard University Press, 1986.

[90] Marcel A Just and Patricia A Carpenter. A Theory of Reading: From Eye Fixations to Comprehension. *Psychological review*, 87(4):329, 1980.

[91] Daniel Kahneman. A Perspective on Judgment and Choice: Mapping Bounded Rationality. *American Psychologist*, 58(9):697, 2003.

[92] Daniel Kahneman and Richard H. Thaler. Anomalies: Utility Maximization and Experienced Utility. *Journal of Economic Perspectives*, 20(1):221–234, March 2006. doi: 10.1257/089533006776526076. URL http://www.aeaweb.org/articles?id=10.1257/089533006776526076.

[93] Stephen Cole Kleene. Introduction to metamathematics. 1954.

[94] KnowBe4. Kevin Mitnick Security Awareness Training. https://www.knowbe4.com/products/kevin-mitnick-security-awareness-training/, 2019.

[95] KnowBe4. The Inside Man: Fake Life. Real Consequences. https://info.knowbe4.com/inside-man-chn, 2019.

[96] Marios Kokkodis. Learning from Positive and Unlabeled Amazon Reviews: Towards Identifying Trustworthy Reviewers. In *Proceedings of the 21st International Conference on World Wide Web*, pages 545–546. ACM, 2012.

[97] Spyros Kokolakis. Privacy Attitudes and Privacy Behaviour: A Review of Current Research on the Privacy Paradox Phenomenon. *Computers & Security*, 64:122–134, 2017.

[98] Peter König, Niklas Wilming, Tim C Kietzmann, Jose P Ossandón, Selim Onat, Benedikt V Ehinger, Ricardo R Gameiro, and Kai Kaspar. Eye movements as a window to cognitive processes. *Journal of Eye Movement Research*, 9(5):1–16, 2016.

[99] Bruno Korbar, Jim Blythe, Ross Koppel, Vijay Kothari, and Sean Smith. Validating an agent-based model of human password behavior. In *AAAI Publications, Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[100] Vijay Kothari, Jim Blythe, Sean Smith, and Ross Koppel. Agent-based modeling of user circumvention of security. In *Proceedings of the 1st International Workshop on Agents and CyberSecurity*, page 5. ACM, 2014.

[101] Vijay Kothari, Jim Blythe, Sean W Smith, and Ross Koppel. Measuring the Security Impacts of Password Policies Using Cognitive Behavioral Agent-Based

Modeling. In *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, page 13. ACM, 2015.

[102] Vijay Kothari, Jim Blythe, Ross Koppel, and Sean Smith. Password Logbooks and What Their Amazon Reviews Reveal About Their Users' Motivations, Beliefs, and Behaviors. In *2nd European Workshop on Usable Security (EuroUSEC 2017)*. IEEE, 2017.

[103] Vijay Kothari, Jim Blythe, Sean Smith, and Ross Koppel. Data Privacy and the Elusive Goal of Empowering the User. In *Workshop on Moving Beyond a 'One-Size-Fits-All' Approach: Exploring Individual Differences in Privacy, CHI*, 2018.

[104] Vijay Kothari, Prashant Anantharaman, Ira Ray Jenkins, Michael C. Millian, J. Peter Brady, Sameed Ali, Sergey Bratus, Jim Blythe, Ross Koppel, and Sean W. Smith. Human-Computability Boundaries. In *Security Protocols Workshop XXVII (To Appear)*. Springer International Publishing, 2020.

[105] Ponnurangam Kumaraguru, Yong Rhee, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. Protecting People from Phishing: The Design and Evaluation of an Embedded Training Email System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 905–914, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240760. URL `http://doi.acm.org/10.1145/1240624.1240760`.

[106] Ponnurangam Kumaraguru, Justin Cranshaw, Alessandro Acquisti, Lorrie Cranor, Jason Hong, Mary Ann Blair, and Theodore Pham. School of Phish: A Real-world Evaluation of Anti-phishing Training. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 3:1–3:12, New York,

NY, USA, 2009. ACM. ISBN 978-1-60558-736-3. doi: 10.1145/1572532.1572536. URL `http://doi.acm.org/10.1145/1572532.1572536`.

[107] Andrew L. Kun, Oskar Palinko, Zeljko Medenica, and Peter Heeman. On the Feasibility of Using Pupil Diameter to Estimate Cognitive Load Changes for In-Vehicle Spoken Dialogues. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 3766–3770. International Speech and Communication Association, 2013.

[108] Meng-Lung Lai, Meng-Jung Tsai, Fang-Ying Yang, Chung-Yuan Hsu, Tzu-Chien Liu, Silvia Wen-Yu Lee, Min-Hsien Lee, Guo-Li Chiou, Jyh-Chong Liang, and Chin-Chung Tsai. A review of using eye-tracking technology in exploring learning from 2000 to 2012. *Educational Research Review*, 10:90 – 115, 2013. ISSN 1747-938X. doi: 10.1016/j.edurev.2013.10.001. URL `http://www.sciencedirect.com/science/article/pii/S1747938X13000316`.

[109] John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An Architecture for General Intelligence. *Artificial intelligence*, 33(1):1–64, 1987.

[110] Butler Lampson. Usable security: How to get it. *Communications of the ACM*, 52(11):25–27, 2009.

[111] Edith Law and Luis von Ahn. Defining (Human) Computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–121, 2011.

[112] Kevin Lewis, Jason Kaufman, and Nicholas Christakis. The taste for privacy: An analysis of college student privacy settings in an online social network. *Journal of Computer-Mediated Communication*, 14(1):79–100, 2008.

[113] Malware-Traffic-Analysis.net. Malware-Traffic-Analysis.net: A source for pcap files and malware samples... `https://malware-traffic-analysis.net`.

[114] Max-Emanuel Maurer, Alexander De Luca, and Sylvia Kempe. Using Data Type Based Security Alert Dialogs to Raise Online Security Awareness. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, SOUPS '11, pages 2:1–2:13, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0911-0. doi: 10.1145/2078827.2078830. URL http://doi.acm.org/10.1145/2078827.2078830.

[115] Aleecia M McDonald and Lorrie Faith Cranor. The cost of reading privacy policies. *ISJLP*, 4:543, 2008.

[116] Robert McMillan. The Man Who Wrote Those Password Rules Has a New Tip: N3v$rM1-d. *The Wall Street Journal*, 8 2017. https://www.wsj.com/articles/the-man-who-wrote-those-password-rules-has-a-new-tip-n3v-r-m1-d-1502124118.

[117] Simon Meier, Benedikt Schmidt, Cas Cremers, and David Basin. The TAMARIN Prover for the Symbolic Analysis of Security Protocols. In *International Conference on Computer Aided Verification*, pages 696–701. Springer, 2013.

[118] Microsoft. Office 365 ATP Safe Links. May 2019.

[119] Caitlin Mills, Jennifer Wu, and Sidney D'Mello. Being Sad Is Not Always Bad: The Influence of Affect on Expository Text Comprehension. *Discourse Processes*, 56(2):99–116, 2019. doi: 10.1080/0163853X.2017.1381059. URL https://doi.org/10.1080/0163853X.2017.1381059.

[120] D. Miyamoto, T. Iimura, G. Blanc, H. Tazaki, and Y. Kadobayashi. EyeBit: Eye-Tracking Approach for Enforcing Phishing Prevention Habits. In *2014 Third International Workshop on Building Analysis Datasets and Gathering*

Experience Returns for Security (BADGERS), pages 56–65, Sep. 2014. doi: 10.1109/BADGERS.2014.14.

[121] Saif Mohammad. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, 2018.

[122] Mozilla. How does built-in Phishing and Malware Protection work? `https://support.mozilla.org/en-US/kb/how-does-phishing-and-malware-protection-work`, 2019.

[123] Susan M. Munn, Leanne Stefano, and Jeff B. Pelz. Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding. In *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, APGV '08, pages 33–42, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-981-4. doi: 10.1145/1394281.1394287. URL `http://doi.acm.org/10.1145/1394281.1394287`.

[124] Naked Security, Sophos. Anatomy of a "goto fail" – Apple's SSL bug explained, plus an unofficial patch for OS X! `https://nakedsecurity.sophos.com/2014/02/24/anatomy-of-a-goto-fail-apples-ssl-bug-explained-plus-an-unofficial-patch/`, February 2014. [Online; accessed 3-January-2019].

[125] Yoav Nathaniel. Attack Report: Office 365 Security Hacked Using Google Redirect, September 2017. `https://www.avanan.com/resources/open-redirect-vulnerability`.

[126] Yoav Nathaniel. Attack Report: Office 365 Security Hacked Using

Google Redirect. *Avanan*, 9 2017. https://www.avanan.com/resources/open-redirect-vulnerability.

[127] Peter G Neumann. Risks Digest: Forum on Risks to the Public in Computers and Related Systems. *ACM Committee on Computers and Public Policy*. https://catless.ncl.ac.uk/Risks/.

[128] Christopher Novak, Jim Blythe, Ross Koppel, Vijay Kothari, and Sean Smith. Modeling aggregate security with user agents that employ password memorization techniques. In *Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[129] Marcus Nyström, Richard Andersson, Kenneth Holmqvist, and Joost van de Weijer. The influence of calibration method and eye physiology on eyetracking data quality. *Behavior Research Methods*, 45(1):272–288, Mar 2013. ISSN 1554-3528. doi: 10.3758/s13428-012-0247-4. URL https://doi.org/10.3758/s13428-012-0247-4.

[130] Jonathan A Obar and Anne Oeldorf-Hirsch. The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. 2016.

[131] Standford Encyclopedia of Philosophy. Peirce's Theory of Signs. http://plato.stanford.edu/entries/peirce-semiotics/.

[132] Charles Kay Ogden and Ivor Armstrong Richards. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, volume 29. K. Paul, Trench, Trubner & Company, Limited, 1923.

[133] Charles Kay Ogden and Ivor Armstrong Richards. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Harcourt, Brace and Company, 1927. ISBN 0156584468.

[134] Gunter Ollmann. The Phishing Guide: Understanding & Preventing Phishing Attacks. *NGS Software Insight Security Research*, 2004.

[135] Kenneth Olmstead and Aaron Smith. Americans and Cybersecurity. *Pew Research Center*, pages 1–5, 2017.

[136] Bruce Schneier. Schneier on Security. Write Down Your Password. `https://www.schneier.com/blog/archives/2005/06/write_down_your.html`. Accessed: 03-11-2017.

[137] OpenDNS. PhishTank. `https://www.phishtank.com`.

[138] Oskar Palinko and Andrew L. Kun. Exploring the Effects of Visual Cognitive Load and Illumination on Pupil Diameter in Driving Simulators. In *Proceedings of the Symposium on Eye Tracking Research & Applications*, ETRA '12, pages 413–416, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1221-9. doi: 10.1145/2168556.2168650. URL `http://doi.acm.org/10.1145/2168556.2168650`.

[139] Oskar Palinko, Andrew L. Kun, Alexander Shyrokov, and Peter Heeman. Estimating Cognitive Load Using Remote Eye Tracking in a Driving Simulator. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pages 141–144, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-994-7. doi: 10.1145/1743666.1743701. URL `http://doi.acm.org/10.1145/1743666.1743701`.

[140] D. Pappusetty, V. V. R. Chinta, and H. Kalva. Using pupillary response to assess video quality. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 64–65, Jan 2017. doi: 10.1109/ICCE.2017.7889231.

[141] Meredith Patterson. Parser combinators for binary formats, in C. https://github.com/UpstandingHackers/hammer. [Online; accessed 4-January-2019].

[142] Bastian Pfleging, Drea K. Fekety, Albrecht Schmidt, and Andrew L. Kun. A Model Relating Pupil Diameter to Mental Workload and Lighting Conditions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5776–5788, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858117. URL http://doi.acm.org/10.1145/2858036.2858117.

[143] PhishingBox. phishingbox. https://www.phishingbox.com, 2019.

[144] PhishLabs. Security Awareness Training. https://www.phishlabs.com/security-awareness-training/, 2019.

[145] Marc Pomplun and Sindhura Sunkara. Pupil Dilation as an Indicator of Cognitive Workload in Human-Computer Interaction. In *Proceedings of the International Conference on HCI*, 2003.

[146] A Poole and Linden Ball. *Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects*, pages 211–219. Idea Group, Inc, 01 2006.

[147] Proofpoint. Proofpoint Security Awareness Training. https://www.proofpoint.com/us/product-family/security-awareness-training, 2019.

[148] Proofpoint. Proofpoint Essentials URL Defense: Advanced Protection with Proofpoint's Targeted Attack Protection. https://www.proofpoint.com/us/resources/data-sheets/essentials-url-defense, 2019.

[149] Florian Quinkert, Martin Degeling, Jim Blythe, and Thorsten Holz. Be the Phisher - Understanding Users' Perception of Malicious Domains. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications (To Appear)*, Taipei, Taiwan, 2020. Association for Computing Machinery.

[150] Alexander J Quinn and Benjamin B Bederson. Human Computation: A Survey and Taxonomy of a Growing Field. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1403–1412. ACM, 2011.

[151] Emilee Rader, Rick Wash, and Brandon Brooks. Stories as Informal Lessons about Security. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 6. ACM, 2012.

[152] Niveta Ramkumar, Nadia Fereydooni, Orit Shaer, and Andrew L. Kun. Visual Behavior During Engagement with Tangible and Virtual Representations of Archaeological Artifacts. In *Proceedings of the 8th ACM International Symposium on Pervasive Displays*, PerDis '19, pages 21:1–21:7, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6751-6. doi: 10.1145/3321335.3324930. URL `http://doi.acm.org/10.1145/3321335.3324930`.

[153] Niveta Ramkumar, Vijay Kothari, Caitlin Mills, Ross Koppel, Jim Blythe, Sean Smith, and Andrew L. Kun. Eyes on URLs: Relating Visual Behavior to Safety Decisions. In *Proceedings of the 12th ACM Symposium on Eye Tracking Research & Applications (To Appear)*, Stuttgart, Germany, 2020. Association for Computing Machinery. doi: 10.1145/3379155.3391328.

[154] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124 3:372–422, 1998.

[155] Keith Rayner and Susan A. Duffy. Lexical complexity and fixation times in

reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & Cognition*, 14(3):191–201, May 1986. ISSN 1532-5946. doi: 10.3758/BF03197692. URL https://doi.org/10.3758/BF03197692.

[156] Keith Rayner and Martin H. Fischer. Mindless reading revisited: Eye movements during reading and scanning are different. *Perception & Psychophysics*, 58(5):734–747, Jul 1996. ISSN 1532-5962. doi: 10.3758/BF03213106. URL https://doi.org/10.3758/BF03213106.

[157] Keith Rayner, Kathryn H Chace, Timothy J Slattery, and Jane Ashby. Eye Movements as Reflections of Comprehension Processes in Reading. *Scientific Studies of Reading*, 10(3):241–255, 2006. doi: 10.1207/s1532799xssr1003_3. URL https://doi.org/10.1207/s1532799xssr1003_3.

[158] Elissa M Redmiles, Amelia R Malone, and Michelle L Mazurek. I Think They're Trying to Tell Me Something: Advice Sources and Selection for Digital Security. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 272–288. IEEE, 2016.

[159] Erik D Reichle, Alexander Pollatsek, and Keith Rayner. Using E-Z Reader to simulate eye movements in nonreading tasks: A unified framework for understanding the eye–mind link. *Psychological Review*, 119(1):155–185, 2012.

[160] Koceilah Rekouche. Early Phishing. 2011.

[161] Karen Renaud and Lewis Mackenzie. Simpass: Quantifying the impact of password behaviours and policy directives on an organisation's systems. *Journal of Artificial Societies and Social Simulation*, 16(3):3, 2013.

[162] RiskyAnalytics. DNS-BH ? Malware Domain Blocklist by RiskAnalytics. http://www.malwaredomains.com.

[163] Dario D. Salvucci and Joseph H. Goldberg. Identifying Fixations and Saccades in Eye-tracking Protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 71–78, New York, NY, USA, 2000. ACM. ISBN 1-58113-280-8. doi: 10.1145/355017.355028. URL http://doi.acm.org/10.1145/355017.355028.

[164] SANS. Robust Phishing Awareness Simulation Training that Changes Behavior. https://www.sans.org/security-awareness-training/products/phishing, 2019.

[165] SANS. The 2019 SANS EndUser Training Suite. https://www.sans.org/security-awareness-training/products/end-user, 2019.

[166] Len Sassaman, Meredith L Patterson, Sergey Bratus, Michael E Locasto, and Anna Shubina. Security Applications of Formal Language Theory. *IEEE Systems Journal*, 7(3):489–500, 2013.

[167] Alexandre Schaefer, Frédéric Nils, Xavier Sanchez, and Pierre Philippot. Assessing the effectiveness of a large database of emotion-eliciting films: A new tool for emotion researchers. *Cognition and Emotion*, 24(7):1153–1172, 2010. doi: 10.1080/02699930903274322. URL https://doi.org/10.1080/02699930903274322.

[168] Emily Schechter. A secure web is here to stay. *Google Security Blog*, 2 2018. https://security.googleblog.com/2018/02/a-secure-web-is-here-to-stay.html.

[169] Shalla Secure Services. Shalla's Blacklists. http://www.shallalist.de.

[170] Oron Shagrir. Gödel on Turing on computability. In A. Olszewski, J. Wole'nski,

and R. Janusz, editors, *Church's Thesis After Seventy Years*, pages 393–419. Ontos Verlag, 2006.

[171] Richard Shay, Abhilasha Bhargav-Spantzel, and Elisa Bertino. Password policy simulation and analysis. In *Proceedings of the 2007 ACM workshop on Digital identity management*, pages 1–10. ACM, 2007.

[172] Steve Sheng, Bryant Magnien, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. Anti-Phishing Phil: The Design and Evaluation of a Game That Teaches People Not to Fall for Phish. In *Proceedings of the 3rd Symposium on Usable Privacy and Security*, SOUPS '07, pages 88–99, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-801-5. doi: 10.1145/1280680.1280692. URL `http://doi.acm.org/10.1145/1280680.1280692`.

[173] Steve Sheng, Mandy Holbrook, Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Julie Downs. Who Falls for Phish?: A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 373–382, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753383. URL `http://doi.acm.org/10.1145/1753326.1753383`.

[174] John L. Sibert, Mehmet Gokturk, and Robert A. Lavine. The Reading Assistant: Eye Gaze Triggered Auditory Prompting for Reading Remediation. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 101–107, New York, NY, USA, 2000. ACM. ISBN 1-58113-212-3. doi: 10.1145/354401.354418. URL `http://doi.acm.org/10.1145/354401.354418`.

[175] Scout Sinclair. *Access Control In and For the Real World*. PhD thesis, Dartmouth College, 2013.

[176] Michael Sipser. *Introduction to the Theory of Computation (Second Edition)*. Thomson Course Technology, 2006.

[177] Steven A Sloman. Two Systems of Reasoning. In Thomas Gilovich, Dale Griffin, and Daniel Kahneman, editors, *Heuristics and Biases: The Psychology of Intuitive Judgment*. Cambridge University Press, 2002.

[178] Sean W Smith. Humans in the loop: Human-computer interaction and security. *IEEE Security & privacy*, 99(3):75–79, 2003.

[179] Sean W Smith. Security and Cognitive Bias: Exploring the Role of the Mind. *IEEE Security & Privacy*, 10(5):75–78, 2012.

[180] Sean W Smith, Ross Koppel, Jim Blythe, and Vijay Kothari. Mismorphism: a Semiotic Model of Computer Security Circumvention. Technical report, Dartmouth College, Department of Computer Science, 03 2015.

[181] Robert F. Stanners, Michelle Coulter, Allen W. Sweet, and Philip Murphy. The Pupillary Response as an Indicator of Arousal and Cognition. *Motivation and Emotion*, 3(4):319–340, Dec 1979. ISSN 1573-6644. doi: 10.1007/BF00994048. URL `https://doi.org/10.1007/BF00994048`.

[182] Keith E Stanovich and Richard F West. Advancing the rationality debate. *Behavioral and brain sciences*, 23(05):701–717, 2000.

[183] Elizabeth Stobert and Robert Biddle. The Password Life Cycle: User Behaviour in Managing Passwords. In *Symposium On Usable Privacy and Security*

(SOUPS 2014), pages 243–255. USENIX Association, 2014. ISBN 978-1-931971-13-3. URL `https://www.usenix.org/conference/soups2014/proceedings/presentation/stobert`.

[184] Simon Stockhardt, Benjamin Reinheimer, Melanie Volkamer, Peter Mayer, Alexandra Kunz, Philipp Rack, and Daniel Lehmann. Teaching Phishing-Security: Which Way is Best? In Jaap-Henk Hoepman and Stefan Katzenbeisser, editors, *ICT Systems Security and Privacy Protection*, pages 135–149, Cham, 2016. Springer International Publishing. ISBN 978-3-319-33630-5.

[185] Mary Frances Theofanos and Shari Lawrence Pfleeger. Guest editors' introduction: Shouldn't all security be usable? *IEEE Security & Privacy*, 9(2):12–17, 2011.

[186] Chuan Annie Tian and Matthew L Jensen. Effects of emotional appeals on phishing susceptibility. In *Proceedings of the 14th Pre-ICIS Workshop on Information Security and Privacy*, volume 1, 2019.

[187] Harshal Tupsamudre, Ajeet Kumar Singh, and Sachin Lodha. Everything Is in the Name – A URL Based Approach for Phishing Detection. In *Cyber Security Cryptography and Machine Learning*, pages 231–248, Cham, 2019. Springer International Publishing.

[188] Alan Mathison Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(1): 230–265, 1937.

[189] Joseph Turow, Michael Hennessy, and Nora A Draper. The Tradeoff Fallacy: How Marketers are Misrepresenting American Consumers and Opening Them Up to Exploitation. *Available at SSRN 2820060*, June 2015.

[190] Jennifer Urban and Chris Hoofnagle. The Privacy Pragmatic as Privacy Vulnerable. In *Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association Berkeley, CA, 2014.

[191] Luis Von Ahn. Human Computation. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 1–2. IEEE Computer Society, 2008.

[192] wanderingbilby. Attack Report: Office 365 Security Hacked Using Google Redirect. *Reddit*. `https://www.reddit.com/r/sysadmin/comments/d9ndnf/heres_a_phishing_url_to_give_you_nightmares/`.

[193] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. Understanding password choices: How frequently entered passwords are re-used across websites. In *Symposium on Usable Privacy and Security (SOUPS)*, 2016.

[194] George RS Weir. Meaningful interaction in complex man-machine systems. *Reliability Engineering & System Safety*, 38(1):151–156, 1992.

[195] Gal Weizman. Critical Security Flaw Found in WhatsApp Desktop Platform Allowing Cybercriminals Read From The File System Access. *PerimeterX*, 2 2020. `https://www.perimeterx.com/tech-blog/2020/whatsapp-fs-read-vuln-disclosure/`.

[196] Zikai Alex Wen, Zhiqiu Lin, Rowena Chen, and Erik Andersen. What.Hack: Engaging Anti-Phishing Training Through a Role-playing Phishing Simulation Game. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 108:1–108:12, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300338. URL `http://doi.acm.org/10.1145/3290605.3300338`.

[197] Rainer Westermann, Kordelia Spies, Günter Stahl, and Friedrich W Hesse. Relative effectiveness and validity of mood induction procedures: A meta-analysis. *European Journal of Social Psychology*, 26(4):557–580, 1996.

[198] WHATWG. Url living standard, August 2019. `https://url.spec.whatwg.org`.

[199] Wikipedia contributors. URL — Wikipedia, The Free Encyclopedia, 2019. URL `https://en.wikipedia.org/w/index.php?title=URL&oldid=909233629`. [Online; accessed 11-September-2019].

[200] Wikipedia contributors. Papers, Please — Wikipedia, The Free Encyclopedia. `https://en.wikipedia.org/w/index.php?title=Papers,_Please&oldid=914101946`, 2019. [Online; accessed 18-September-2019].

[201] Wikipedia contributors. Phishing — Wikipedia, The Free Encyclopedia. `https://en.wikipedia.org/w/index.php?title=Phishing&oldid=914413259`, 2019. [Online; accessed 8-September-2019].

[202] Wikipedia contributors. Triangle of reference — Wikipedia, The Free Encyclopedia, 2019. URL `https://en.wikipedia.org/w/index.php?title=Triangle_of_reference&oldid=895514020`. [Online; accessed 23-April-2020].

[203] Wikipedia contributors. URL redirection — Wikipedia, The Free Encyclopedia. `https://en.wikipedia.org/w/index.php?title=URL_redirection&oldid=916373985`, 2019. [Online; accessed 19-September-2019].

[204] Wikipedia contributors. Cognitive load — Wikipedia, The Free Encyclopedia. `https://en.wikipedia.org/w/index.php?title=Cognitive_load&oldid=952015469`, 2020. [Online; accessed 29-April-2020].

[205] Wikipedia contributors. Grounded theory — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Grounded_theory&oldid=950547616, 2020. [Online; accessed 27-April-2020].

[206] Wikipedia contributors. IDN homograph attack — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&oldid=948172087, 2020. [Online; accessed 29-April-2020].

[207] Wikipedia contributors. Minimum wage in the United States — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Minimum_wage_in_the_United_States&oldid=953540283, 2020. [Online; accessed 28-April-2020].

[208] Wikipedia contributors. Monospaced font — Wikipedia, The Free Encyclopedia, 2020. URL https://en.wikipedia.org/w/index.php?title=Monospaced_font&oldid=949685729. [Online; accessed 28-April-2020].

[209] Wikipedia contributors. Qualtrics — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Qualtrics&oldid=952250584, 2020. [Online; accessed 29-April-2020].

[210] Wikipedia contributors. Semiotics — Wikipedia, The Free Encyclopedia, 2020. URL https://en.wikipedia.org/w/index.php?title=Semiotics&oldid=951660081. [Online; accessed 23-April-2020].

[211] Wikipedia contributors. Time-of-check to time-of-use — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Time-of-check_to_time-of-use&oldid=944275708, 2020. [Online; accessed 29-April-2020].

[212] Wikipedia contributors. URL shortening — Wikipedia, The Free Encyclopedia, 2020. URL https://en.wikipedia.org/w/index.php?title=URL_shortening&oldid=953486745. [Online; accessed 28-April-2020].

[213] Jeannette M Wing. Computational Thinking. *Communications of the ACM*, 49(3):33–35, 2006.

[214] Min Wu, Robert C. Miller, and Simson L. Garfinkel. Do Security Toolbars Actually Prevent Phishing Attacks? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 601–610, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124863. URL http://doi.acm.org/10.1145/1124772.1124863.

[215] Aiping Xiong, Robert W Proctor, Weining Yang, and Ninghui Li. Is domain highlighting actually helpful in identifying phishing web pages? *Human factors*, 59(4):640–660, 2017.

[216] Jie Xu, Yang Wang, Fang Chen, Ho Choi, Guanzhong Li, Siyuan Chen, and Sazzad Hussain. Pupillary Response Based Cognitive Workload Index Under Luminance and Emotional Changes. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1627–1632, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0268-5. doi: 10.1145/1979742.1979819. URL http://doi.acm.org/10.1145/1979742.1979819.

[217] Ka-Ping Yee. Guidelines and strategies for secure interaction design. *Security and Usability: Designing Secure Systems That People Can Use*, 247, 2005.

[218] Johannes Zagermann, Ulrike Pfeil, and Harald Reiterer. Measuring Cognitive Load Using Eye Tracking Technology in Visual Computing. In *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods*

for Visualization, BELIV '16, pages 78–85, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4818-8. doi: 10.1145/2993901.2993908. URL `http://doi.acm.org/10.1145/2993901.2993908`.