

ABSTRACT

NEIL, LORENZO COLIN. Understanding Root Causes for Online Cybersecurity Advice Challenges. (Under the direction of Dr. Bradley Reaves).

Security advice is constructed with the intention of assisting everyday users of technology on how to manage the security of their computers and devices. While there is an abundance of security advice available, we do not know the overall quality of the advice. Prioritization of security advice is also an issue as both end-users and experts can not agree on which advice is most important. Challenges in advice quality and prioritization ultimately makes using security advice much more difficult for end-users.

This dissertation addresses challenges in security advice coverage, content creation, and advice usability. Our first work investigates the completeness of security advice for account remediation from popular web services. We perform this analysis by creating a model for the account remediation process, that we use to qualitatively code the advice. Account remediation is a systematic process that includes important security practices that all end-users need to be secure. Our findings in this work indicate that account remediation advice from web services studied is insufficient in coverage, and does not fully address all phases for account remediation.

In our second work, we focus on a broader scope of advice and look into how general security advice is written and prioritized by security advice writers. We interviewed 21 authors of security advice from multiple different organizations and identified key processes and decision making motivations considered among advice writers during the writing process. We also identified common challenges reported by advice writers that contribute to an overproduction of security advice topics and haphazard advice construction.

Finally, we conclude this dissertation with exploring challenges that new users face when using secret management tools (SMT), with a focus on SMT tool documentation, to manage secrets. In a mixed-methods study, we explore in depth the challenges and experiences users report when using SMTs to perform secret management tasks. We observe participants in person use SMTs, with the assistance of available tool documentation, to perform two secret management tasks: Secret storage and access, then secret injection. We find that the quality of available tool documentation for SMTs drastically impacts a user's ability to effectively use the tools, which ultimately affects users abilities to manage secrets.

© Copyright 2025 by Lorenzo Colin Neil

All Rights Reserved

Understanding Root Causes for Online Cybersecurity Advice Challenges

by
Lorenzo Colin Neil

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina
2025

APPROVED BY:

Dr. Anupam Das

Dr. Laurie Williams

Dr. Sandeep Kuttal

Dr. Yasemin Acar

Dr. Bradley Reaves
Chair of Advisory Committee

DEDICATION

To my family.

BIOGRAPHY

Lorenzo C. Neil was born in Trenton, New Jersey and raised in Silver Spring, Maryland. He received his Bachelor of Science in Computer Science from the University of Maryland, Baltimore County in May 2019. In Fall 2019, Lorenzo joined the doctoral program at North Carolina State University, as well as the Wolfpack Security and Privacy Research Lab. During his studies, he authored three papers as the first author, co-authored two additional papers, and was a NIST Graduate Student Measurement Science and Engineering Fellow in the Human-Centered Cybersecurity Program. At the time of this dissertation document, he also two more first author papers in submission. After completing his doctoral degree, he joined the Institute for Defense Analyses as a full-time Research Staff Member.

ACKNOWLEDGEMENTS

I want to thank my advisor, Dr. Bradley Reaves, and my external collaborator, Dr. Yasemin Acar, for their guidance and support during my program at North Carolina State University. Dr. Reaves introduced me to the academic field of security research and taught me how to excel academically and professionally as a researcher. Dr. Acar introduced me to usable security research and has been instrumental towards my career in developing qualitative research methods. The work in this dissertation was only possible with their consistent guidance and support.

Thank you to my Wolfpack Security and Privacy Research Lab (WSPR) colleagues. A special thank you goes to Setu Kumar Basak for working with me on two of his papers investigating developer secret management practices. Second, I am thankful to former WSPR colleagues Elijah Bouma-Sims and Evan Lafontaine for being co-authors on my first full paper during my PhD which investigates the coverage of account remediation advice. Third, I am thankful to my external collaborators Harshini Sri Ramulu and Deepthi Mungara at Paderborn University who were co-authors to my papers discussed in Chapters 3 and 5, respectively. Fourth, I am grateful to Dr. Julie Haney from NIST for granting me the opportunity to work with her and other NIST researchers on multiple projects, as well as providing professional and academic guidance. Fifth, I would also like to thank my doctoral committee members: Dr. Anupam Das and Dr. Laurie Williams, and Dr. Sandeep Kuttal for their invaluable feedback and discussions throughout my degree.

I want to also thank my family who has provided support along the way. My parents, Larold and Faustina, have been professional role models and provided loving support my whole life. My sister, Christina, has also provided support for my PhD journey as well. I want to thank my surrounding family as well. Lastly, I want to thank my partner, Amber Meeks, and her family for being supportive through my journey while in North Carolina.

The research comprising this dissertation was supported in part by the National Science Foundation under Award No. CNS-2055554 and by the 2020 USENIX IDP Award Funds. Any findings and opinions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

TABLE OF CONTENTS

List of Tables	viii
List of Figures	ix
Chapter 1 Introduction	1
1.1 Area	1
1.2 Problem	2
1.3 Thesis Statement	3
1.4 Contributions	4
1.5 Organization	4
Chapter 2 Related Work	6
2.1 Security Mental Models	6
2.2 Account Compromise	7
2.3 Account Recovery	7
2.4 Sources for Security Advice	8
2.5 Gaps in Security Advice Implementation	8
2.6 Advice Analysis	9
2.7 Advice Prioritization	9
2.8 Secret Management	10
2.9 Secret Management Tools	11
2.10 Software Tool Documentation	11
2.11 Security Tool Usability and Adoption	13
Chapter 3 Investigating Web Service Account Remediation Advice	14
3.1 Introduction	14
3.2 Methods	16
3.2.1 Codebook Development	16
3.2.2 Account Remediation Model	17
3.2.3 Training and Reliability	18
3.2.4 Collecting Advice from Web Services	18
3.2.5 Differences in Coverage of Account Remediation Advice	21
3.2.6 Limitations	23
3.3 Account Remediation Advice Coverage Results	24
3.3.1 Overall Phase Coverage	25
3.3.2 Content Analysis by Phase	26
3.3.3 85% of Web Services did not provide Account Remediation Advice	34
3.3.4 Coverage of Advice versus Popularity	34
3.3.5 Coverage of Advice versus Disclosed Data Breach	35
3.4 Discussion	36

3.5	Conclusion	39
Chapter 4 Who Comes Up with this Stuff? Interviewing Authors to Understand How They Produce Security Advice		
4.1	Introduction	41
4.2	Methods	43
4.2.1	Participant Recruitment	43
4.2.2	Instrument Creation	45
4.2.3	Interview Process	46
4.2.4	Data Protection	46
4.2.5	Data Analysis	46
4.2.6	Limitations	47
4.3	Results	48
4.3.1	Information Gathering	49
4.3.2	Advice Drafting and Decision-Making	51
4.3.3	Collaboration and Review	55
4.3.4	Reported Challenges and Improvements	56
4.4	Discussion	58
4.4.1	Identifying Lack of Consensus	59
4.4.2	Methodological Improvements for Advice Writing	62
4.5	Conclusion	65
Chapter 5 It Should Be Easy but... New Users' Experiences and Challenges with Secret Management Tools		
5.1	Introduction	66
5.2	Methodology	68
5.2.1	Recruitment and Eligibility Criteria	69
5.2.2	Study Design	71
5.2.3	Post-Task Interviews	77
5.2.4	Data Protection/Ethical Standards	78
5.2.5	Limitations	79
5.3	Results	80
5.3.1	Participant Performances	80
5.3.2	Overall Experience and Challenges	83
5.3.3	Tool Documentation Challenges	84
5.3.4	Secondary Sources	88
5.3.5	Tool Usability Feedback	91
5.4	Discussion	92
5.4.1	RQ1: New Users and SMT Usage	92
5.4.2	RQ2: Factors that Impact SMT Usage	93
5.4.3	RQ3: Comparison of Different SMTs	94
5.4.4	Recommendations	95

5.5	Conclusion	96
Chapter 6	Conclusion and Future Work	97
6.1	Account Remediation Support	98
6.2	Secret Management Tools and Documentation	99
References		101
APPENDICES		114
Appendix A	Investigation of Account Remediation Full Codebook and Web Services Studied	115
A.1	Codebook	116
A.2	Web Services Studied	120
A.3	Data	121
Appendix B	Interviews with Security Advice Authors Interview Guide and High Level Codes	122
B.1	Interview Guide	122
B.2	Codebook	125
Appendix C	Mixed-methods Study with New SMT Users Interview Questions and Codebook	127
C.1	Interview Guide	127
C.2	Interview Codebook	130

LIST OF TABLES

Table 4.1	Participant Demographics.	45
Table 5.1	Participant demographics and SMT Assignment.	70
Table 5.2	Participants' Performance in Task 1.	81
Table 5.3	Participants' Performance in Task 2.	82
Table A.1	Codebook for Compromise Discovery codes among web services. . . .	116
Table A.2	Codebook for Account Recovery codes among web services.	117
Table A.3	Codebook for Limiting Access codes among web services.	117
Table A.4	Codebook for Service Restoration codes among web services.	118
Table A.5	Codebook for Prevention codes among web services.	119
Table A.6	List of web services studies for account remediation advice.	121
Table B.1	Full list of high level codes from codebook for security advice interviews.	126
Table C.1	Interview Codebook of Participants' Experiences and Wrap-up. . . .	130
Table C.2	Interview Codebook of Tasks 1 and 2.	131

LIST OF FIGURES

Figure 3.1	Methodology: codebook and model development, data collection and analysis.	16
Figure 3.2	Bar graph of all account remediation phases among web services. Limiting Access advice is mentioned in less than half of the web services we analyzed. Service Restoration advice was mentioned in 74% of the web services. All other phases were mentioned by at least 90% of the web services we analyzed.	25
Figure 3.3	How many web services mentioned at least n amount of web services, where n is either at least 5,4,3,2,or 1 phase. Only 39% web services gave advice for all five phases.	26
Figure 3.4	Bar graph of Compromise Discovery codes among web services. Unauthorized or suspicious activity was the highest covered code with 39 web services. No other code was mentioned in more than half of the web services.	28
Figure 3.5	Bar graph of Account Recovery codes among web services. Password reset was mentioned by 91% of services and customer service support was mentioned by 63% of services.	30
Figure 3.6	Bar graph of Limiting Access codes among web services. No single code was mentioned in more than a third of the web services.	31
Figure 3.7	Bar graph of Service Restoration codes among web services. No single code was mentioned in more than 42% of web services.	32
Figure 3.8	Bar graph of Prevention codes among web services. Four out of 11 codes were mentioned in at least 60% of web services and strong password advice was mentioned in 88% of web services.	33
Figure 3.9	Graph of the mean number of phases covered in account remediation by all experimental groups. <i>Very popular</i> web services had a higher mean count of phases mentioned in their account remediation advice than <i>less popular</i> web services. <i>Data breach disclosed</i> services had a higher mean count of phases mentioned in their account remediation advice than <i>non-data breach disclosed</i> web services.	35
Figure 4.1	How experts write general security advice.	48
Figure 5.1	Screenshot image of the Python file, <code>SecureSecret.py</code> , used by participants for Task 1 and 2.	73
Figure 5.2	Screenshot image showing the HCP Vault Secrets dashboard for the <i>Company Secret Storage</i> project with <code>CompanySecretToken</code> stored.	74
Figure 5.3	Sample presentation of the list of CLI commands required to complete Task 1 for HCP Vault Secrets.	74

Figure 5.4 Sample presentation of the injection run CLI command required to complete Task 2 for HCP Vault Secrets. 75

Figure 5.5 Additional part of HashiCorp vault secrets documentation in their web dashboard that reflects the required CLI command for Task 2. . 76

CHAPTER

1

INTRODUCTION

1.1 Area

Computer end-users receive and have access to an abundance of security information from many different sources. Such sources include awareness workshops, games, media, and most commonly from web pages found online that provide security advice. Online security advice is written with the intention of assisting end-users in implementing secure practices. Most security advice that is publicly available, online or written in text, is intended to be broadly applicable towards a wide audience.

Security incidents serve as huge motivations for security advice construction as security advice writers prioritize their content on widely known security incidents. Examples of such security incidents that are widely covered in security advice consists of phishing email attacks, multi-factor authentication, password security, or account compromise remediation. End-users who store information online and access web services refer to online security advice for assistance in protection against security incidents. Even end-users who possess fundamental security knowledge refer to technical or security documentation

to learn how to securely use tools to manage sensitive information.

1.2 Problem

The goal of online security advice is to inform people on how to manage the security of their technology, yet prior research efforts have learned that security advice has several issues. First, we receive so much security advice to the point where we do not know which security advice we should prioritize (Redmiles et al. 2016a,b; Herley 2009; Nicholson et al. 2019; Herley 2013). Prior research has identified that both security experts and non-expert end-users have difficulties in prioritizing the vast amount of security advice available online (Redmiles et al. 2020; Nicholson et al. 2019). This makes the task of even creating security advice difficult if security advice writers cannot agree or understand how they should prioritize content for security advice. Thus, adding more burden onto end-users who rely on security advice and need to make their own decisions on which advice to adopt. If end-users are unable to use advice effectively, their security and privacy remain compromised.

After end-users go through the challenge of finding security advice they believe is relevant, the quality of that advice is also of paramount importance. While there is much information on widely covered security incidents (e.g. password security), it is uncertain if there is enough information for security incidents that are less covered, but still important for the security of end-users. For example, account compromise remediation is a technically complex process with multiple steps. Web services that allow for end-user account creation should provide remediation advice that covers all necessary steps to complete remediation. However, it is unclear if the advice is complete in coverage. It is also unclear how many websites that allow users to create accounts even provide users account remediation advice at all.

While security advice coverage and prioritization are major challenges affecting security advice, end-users also have difficulty rationalizing which advice to adopt (Herley 2009; Fagan and Khan 2016; Christin et al. 2011; Busse et al. 2019) and even finding helpful sources for security advice (Turner et al. (2021); Rader and Wash (2015); Rader et al. (2012); Nicholson et al. (2019)). With all of these known challenges that affect security advice, it begs the question as to whether end-users can effectively use security advice. This question of whether end-users can effectively use security advice applies to everyone, even expert

end-users who possess fundamental security or technical knowledge. For example, expert end-users such as software developers rely on online software development documentation for guidance on how to write secure code or use tools to secure their applications (Dang-Pham et al. 2017; Acar et al. 2016, 2017b; Treude and Robillard 2016; Arya et al. 2023). If the online documentation that software developers have access to is not effective, then developers cannot effectively use tools or implement secure applications. If experts such as software developers lack sufficient guidance on how to effectively use tools or application, then they also may lack the ability to securely handle sensitive information, which can result in sensitive information being insecurely managed or even leaked (Mcdaniel 2023b; Abrams 2023; Meli et al. 2019; Burgees 2024; Jackson 2023).

1.3 Thesis Statement

In this thesis statement, we explain our research in addressing challenges in security advice coverage, content creation, and usability. We present our findings from investigating the quality of account remediation advice, analyzing the construction of general security advice, and exploring the challenges and new experiences new end-users face when using software secret management tools to manage secrets. In our first study, we investigated account remediation advice from 57 popular U.S.-based web services. In the same study, we identified five key phases of account remediation, used this five-phase model to develop a codebook of account remediation advice, then analyzed topic coverage. In our second study, we conducted semi-structured interviews with 21 authors of online security advice to understand: The processes experts perform when writing general security advice, questions experts consider during the advice writing process, and challenges they face when writing general security advice. Finally, we ran a mixed-methods laboratory study with 21 Computer Science (CS) Master's students with prior academic or professional experience with managing secrets. We explored in depth the challenges and experiences they reported when using secret management tools, with a specific focus on tool documentation, to securely manage software secrets. Our final study observed how the quality of tool documentation impacted users' abilities to use secret management tools to manage secrets in our study. Using the findings from all three of our papers investigating challenges with security advice, we give the following thesis statement:

Thesis Statement: *Challenges with online security advice usage are caused by insufficient*

coverage, reactive creation, and haphazard overproduction.

1.4 Contributions

This thesis proposal makes the following contributions:

1. **Analyze the coverage of account remediation advice among web services:** We analyze the coverage of publicly available account remediation advice from popular U.S.-based web services. We perform this analysis through the creation of a systematic five-phase model for account remediation. We characterize the coverage of relevant phases for account remediation in web service advice.
2. **Identify writing processes, key decision making, and challenges among security advice authors:** We identify a model for the advice writing process and address the decision-making that experts consider when writing general security advice. We also identify challenges advice writers face during advice creation, such as retrieving relevant information for their content and writing advice that is suitable across different implementations for a broad audience.
3. **Identify a lack of consensus in security advice writing:** We identify themes for a lack of consensus for procedural decision-making, perceived responsibilities by advice writers, and advice content prioritization.
4. **Explore challenges with using secret management tools, tool documentation, and tool interfaces:** We explore specific challenges with tool documentation and tool interfaces that impact new users' abilities to effectively use secret management tools to manage secrets.

1.5 Organization

The remainder of this thesis proposal is organized as follows:

Chapter 2 provided information on prior related work for account remediation and general security advice. Chapter 3 describes how we create a model for the account remediation process and use it to investigate account remediation advice from popular U.S- web services. Chapter 4 explains our interview study with security advice authors and analyze

the processes, key decision making, and challenges that impact general security advice construction. Chapter 5 describes our mixed-methods study which explores challenges and experiences new users face when using secret management tools to manage secrets. Finally, in Chapter 6, we conclude this dissertation and propose future future directions for future research.

CHAPTER

2

RELATED WORK

Our work relates to multiple areas within security advice: advice coverage, advice content creation, and advice usability. Our work also relates to developer secret management tools, with a specific focus in secret management tool documentation.

2.1 Security Mental Models

A user's mental model on security ultimately informs their security decisions with their devices and online services (Fulton et al. 2019a). Prior research has focused on the user's security mental model (Redmiles et al. 2016b; Bravo-Lillo et al. 2010) and how they interpret security advice and warnings (Akhawe and Felt 2013a). Improving a user's basic knowledge in security limits the chances of their online services being compromised (Bravo-Lillo et al. 2010), though users may reject the advice if it presents a poor cost-to-benefit ratio or it threatens their privacy (Redmiles et al. 2016b,b; Herley 2009; Redmiles et al. 2020).

Inaccurate security mental models impact security decisions made within both the general public and expert computing communities (Bravo-Lillo et al. 2010; Xie et al. 2011).

Redmiles et al. suggest that user security decisions can be modeled as a function of past behavior and knowledge of costs, risks, and context of potential security decisions (Redmiles et al. 2018). Understanding the prior work in how security knowledge is given to users, we look to investigate the processes that generate general security advice. We focus on how general security advice is constructed in order to bridge a gap between the writing processes used and user security decision making that has been reported in prior work.

2.2 Account Compromise

Many account compromises stem from stolen credentials. Prior work has measured how the risk of stolen credentials varies between phishing, malware, or data breaches and predicts the chances for total online account takeover from stolen credentials (Thomas et al. 2017; Peng et al. 2019; Onaolapo et al. 2016). Billions of stolen usernames and passwords are also widely available in underground forums (Thomas et al. 2019; Pal et al. 2019; Wang and Reiter 2020; VanDam et al. 2017); these data sets have been used to create systems that alert users if their usernames or passwords are vulnerable and have been publicly exposed (Thomas et al. 2019; Pal et al. 2019; Wang and Reiter 2020). Other work on detecting compromised accounts (Shay et al. 2014) focused on building models to represent normal account behavior and then using that behavior to analyze current account behavior for anomalies or unusual activity (Egele et al. 2015; Ruan et al. 2015; Cidon et al. 2019; Karimi et al. 2018). Recent work has investigated whether users are informed about data breaches, how they feel about them, and whether they have taken or plan on taking action (Mayer et al. 2021). In our work, we go beyond compromise discovery and account recovery, also focusing on remediating harm to the compromised accounts.

2.3 Account Recovery

Account recovery mechanisms restore access to an account after credentials are lost or changed by an attacker after a compromise. Virtually all widely used password recovery mechanisms, including secret questions and e-mail reset links, have well-understood vulnerabilities and deployment limitations (Parkin et al. 2015). Many major webmail providers employ security questions that can be solved through data mining, are easily guessable, or have low memorability over time (Schechter et al. 2009; Bonneau et al. 2015). Prior work on

account recovery mechanisms investigated different authentication schemes (Bonneau et al. 2012) and password reset strategies (Huh et al. 2017). Password recovery schemes may also be vulnerable to man-in-the-middle (MitM) attacks (Gelernter et al. 2017; Guri et al. 2016). User advice can cover all five phases of account remediation, though a significant body of work has focused on detecting compromise and account recovery. Compromise detection and account recovery have both been widely studied topics, yet to the best of our knowledge, we are the first to study account *remediation* from a holistic perspective.

2.4 Sources for Security Advice

Non-expert computer users regularly make security-related decisions based on informal sources of information such as media (Fulton et al. 2019b; Nicholson et al. 2019), stories from peers (Rader et al. 2012), and web pages containing computer security advice (Redmiles et al. 2016a; Herley 2013; Turner et al. 2021). The resulting impact of these informal sources acting as a source for general user security advice can produce an incomplete understanding in accurately perceiving security threats (Rader and Wash 2015; Boyd et al. 2021). Formal sources of information that provide general security advice consist of security games (Denning et al. 2013; Sheng et al. 2007), training programs (Kumaraguru et al. 2009; Williams et al. 2010; Kumaraguru 2009; Kumaraguru et al. 2009; Sheng et al. 2007), and literature (Mekhail et al. 2014; Zhang-Kennedy et al. 2016). The sources in which users retrieve insight general security impact their security mental models and then ultimately their decision making (Redmiles et al. 2016b, 2017; Nicholson et al. 2019).

2.5 Gaps in Security Advice Implementation

Both the general public and technical experts consider the tradeoffs of implementing general security advice (Redmiles et al. 2020; Busse et al. 2019; Reeder et al. 2017; Haney et al. 2018; Dang-Pham et al. 2017; Beautement et al. 2008) and have conflicting perceived responsibilities as to who is responsible for security advice implementations (Haney et al. 2021; Xie et al. 2011; Kanniah and Mahrin 2016). For example, Haney et al. found that smart home device users have perceived breakdowns in the relationship among who is responsible for the security of their devices between consumers, manufacturers, and relevant third parties such as the government (Haney et al. 2021). We build from this prior work to

investigate who is involved in writing general security advice and how do advice writers determine their perceived set of responsibilities. Our paper aims to understand the decision making that advice writers perform during the writing process, so that we can analyze how the writers themselves perceive and create general security advice.

2.6 Advice Analysis

In recent years, many researchers have analyzed the quality of security advice for both expert (Acar et al. 2016, 2017b; Gorski et al. 2018) and non-expert (Redmiles et al. 2020; Neil et al. 2021, 2020; Akhawe and Felt 2013b) computer users. Prior work from Acar et al. evaluated the state of security practices from popular web resources that developers use within their programming (Acar et al. 2016, 2017b). They found a prevalence of security bugs within current guidance systems, therefore identifying insecure programming practices being advised to developers who seek these web resources (Acar et al. 2016, 2017b).

2.7 Advice Prioritization

Since formal security guidance is not as widespread, online general security advice has been crafted to help users practice secure online habits. However, the general public is supplied with an overabundance of security advice and therefore has to prioritize which advice they will follow (Redmiles et al. 2016a,b; Herley 2009; Nicholson et al. 2019; Herley 2013). Prior work has suggested that users typically perform a cost-to-benefit analysis to determine if the benefits of the advice found are worth the cost of implementing the advice (Herley 2009; Fagan and Khan 2016; Christin et al. 2011; Beautement et al. 2008). Herley et al. analyzed the cost-benefit tradeoff through various forms of security advice to determine much of the available security advice offers a poor cost-benefit tradeoff, therefore prompting users to reject advice (Herley 2009).

The work closest to ours is by Redmiles et al. (Redmiles et al. 2020), in which they investigate the quality of security and privacy advice on the web. Their work breaks down the quality evaluation by examining if security advice on the web is comprehensible, actionable, and effective. Their work concludes that the majority of the advice they investigated is perceived as actionable and comprehensible by both users and experts. However, users and experts both failed to come to a consensus as to what specific advice should be prioritized

(Redmiles et al. 2020). Not only did experts consider 89% of the 374 identified pieces of advice to be useful, they also struggled with internal consistency and alignment with the latest security guidelines.

The key challenges in addressing the volume and prioritization of security advice, as identified by Redmiles et al. (Redmiles et al. 2020), serve as motivations for our work. In this paper, we seek to understand what processes are implemented to write general security advice, as well as what decisions are considered when constructing the advice. We also seek to learn the challenges faced during the advice writing process that impacts the advice content. In doing so, we discover how advice writers gather information to draft advice content, how advice content is prioritized by the writers, and how procedural decision making and responsibilities are perceived by the writers.

2.8 Secret Management

In response to secret leakage, recommended guidelines for software secret management from organizations (Chandramouli 2023; ReversingLabs 2023), coding platforms (Mcdaniel 2023a; GitHub 2023b,a), and security blogs (Katz 2023) were created to guide developers on how to secure secrets. Academic research has also addressed secret management leakage and practices. Meli et al. (Meli et al. 2019) conducted studies to characterize widespread secret leakage in public repositories. Similar work has outlined solutions (Sinha et al. 2015; Basak et al. 2022; Tahaei and Vaniea 2019) and proposed datasets (Basak et al. 2023a) to help developers detect and prevent exposed secrets in their repositories. Basak et al. (Basak et al. 2023b) characterized challenges and solutions reported in questions related to checked-in secrets on Stack Exchange, while Krause et al. (Krause et al. 2023) surveyed 109 developers and interviewed 14 developers to learn common challenges with managing and leaking secrets in code repositories.

Both of their works motivate this study, as they highlight that developers want to use SMTs but struggle with adoption, often citing a lack of available documentation as a key challenge (Basak et al. 2023b; Krause et al. 2023). We further this effort by observing how participants use SMTs and the challenges they face in effectively managing secrets in direct relation to available documentation.

2.9 Secret Management Tools

Secret management tools are highly recommended for securely storing sensitive software secrets in a centralized location (Matsiako 2023; g2 2023). SMTs can offer different functionalities or use cases. Cloud-specific SMTs are built on major cloud ecosystems such as AWS (AWS 2023), Google Cloud Platform (Google 2023), or Microsoft Azure (Microsoft 2023), and are typically used by companies already integrated into those ecosystems. On the other hand, Cross-cloud SMTs are usually offered either as Software-as-a-Service (SaaS) or as a managed service that can be self-hosted or managed (Matsiako 2023). Cross-cloud SMTs are not tied to a specific cloud and offer easier onboarding for developers not working within large cloud ecosystems. Open source SMTs are self-hosted by developers, while in-house SMTs are custom-built by companies for their own use. While both give developers much creative freedom, these services must be regularly maintained by the company or developers themselves, therefore requiring significant overhead. Interview findings from Krause et al. (Krause et al. 2023) show that developers report setup and learning constraints when first adopting SMTs. For our work, we investigate new user experiences with three cross-cloud SMTs and respective tool documentation.

2.10 Software Tool Documentation

Much of the prior work towards software documentation specifically investigated challenges with API focused software documentation that prevents developers from effectively and securely using APIs (Uddin and Robillard 2015; Acar et al. 2017b; Sohan et al. 2017; Treude and Robillard 2016; Robillard and Chhetri 2015; Head et al. 2018; Acar et al. 2016; Subramanian et al. 2014; Acar et al. 2017a). Challenges highlighted from prior work that affect the quality of software documentation include, but are not limited to, insufficient coding examples (Aghajani et al. 2020, 2019; Nassif et al. 2022; Acar et al. 2017b,a; Subramanian et al. 2014), incorrect or ambiguous explanations (Uddin and Robillard 2015; Wen et al. 2019; Middleton et al. 2020; Chen and Huang 2009), as well as non intuitive presentation of material (Nassif and Robillard 2023c,a). Such challenges within software documentation make it more difficult for developers to learn new technology (Nassif and Robillard 2023b; Arya et al. 2023; Chen and Huang 2009) and in some cases affect developer productivity (Wagner and Murphy-Hill 2019; Segal 2007; Murphy-Hill et al. 2019; Head et al. 2018; Noda et al. 2023). In 2003, Lethbridge et al. (Lethbridge et al. 2003) interviewed software

engineers to learn how they used software documentation. 61% of participants in their study felt software documentation was most effective when learning new software and that they prefer simplified documentation, while tending to ignore complex or time consuming documentation (Lethbridge et al. 2003). More recent work highlights how developers seek both, official documentation and also other resources such as Stack Overflow (Arya et al. 2023; Baltes et al. 2020; Robinson et al. 2022; Storey et al. 2024; Acar et al. 2016) to learn new technology. Stack Overflow in particular has been highlighted in prior work as a resource developers access to find answers to challenges addressed by their community (Storey et al. 2024; Parnin et al. 2012). Parnin et al. (Parnin et al. 2012) examined crowd contributed API documentation from Stack Overflow and found that while the crowd documentation provided many examples and explanations for API elements, the rate of information produced from the crowd would not be fast enough to outright replace API documentation (Parnin et al. 2012). A lab study conducted by Acar et al. (Acar et al. 2016) observed Android developers writing security-and-privacy-relevant code under time constraints using different information sources. Their findings reported that developers who used Stack Overflow as a resource wrote less secure but more functional code as opposed to developers who used official Android documentation (Acar et al. 2016). While we see the critical need to create effective software documentation, there is a lack of incentive or motivation to create such documentation (Parnin et al. 2012; Arya et al. 2024). Arya et al. (Arya et al. 2024) interviewed 26 volunteer documentation contributors to learn their motivations for contributing to software documentation. Their work concluded that contributors to software documentation were mostly self motivated, including personal experiences with inadequate software documentation or pursuits for content creation.

Challenges with software documentation content, presentation, and availability as highlighted from prior work serve as motivation for this work. We extend the body of literature by observing how challenges for SMT tool documentation affect our participants' abilities to use SMTs to perform tasks involving managing secrets. We also observe how tool documentation content challenges motivate participants to find secondary sources for further assistance or try workaround methods.

2.11 Security Tool Usability and Adoption

The usability of security tools impacts developers' ability to (securely) complete tasks (Storey et al. 2019; Cheng et al. 2022; Acar et al. 2017a; Witschey et al. 2015). Storey et al. (Storey et al. 2019) surveyed software developers at a company to learn about challenges they faced in regards to their job satisfaction and perceived productivity. Challenges with software architecture and finding relevant information were two of the most common challenges their survey participants reported in regards to their job satisfaction and perceived productivity. Acar et al. (Acar et al. 2017a) compared the usage of different cryptographic API usage by Python developers to write secure code and found that simplified libraries within the APIs helped developers produce more secure code than the comprehensive libraries. Related prior work also echoes the theme that simplifying information or the design of tools improves tool usability (Smith et al. 2020; Indela et al. 2016). In response, prior work has suggested new approaches aimed to help developers write more secure code when using cryptographic APIs (Gorski et al. 2018; Krüger et al. 2017). Gorski et al. (Gorski et al. 2018) observed improvements in code security when comparing developers who used Python's PyCrypto API to developers with a version of PyCrypto with integrated security advice. Krüger et al. (Krüger et al. 2017, 2023) introduced CogniCrypt, their proposed tool for helping developers write code and implement tasks in a secure manner.

Much of the prior efforts on examining security tool usability and adoption focused on APIs or other specific areas. We use prior work as an influence to shape how we investigate the usability of SMTs, which has only been briefly discussed in prior work (Krause et al. 2023; Basak et al. 2023b). We also examine how provided tool documentation and the availability of secondary sources impact the ability of participants to use SMTs for managing secrets.

CHAPTER

3

INVESTIGATING WEB SERVICE ACCOUNT REMEDICATION ADVICE

3.1 Introduction

Online web services allow people to create accounts that store information and communicate with others. Compromises of these accounts are a pervasive problem, with billions of accounts being compromised in 2019 alone (Leonhardt 2019). Account compromises allow the attacker to steal service, surveil the activities of the victim, abuse the system, or otherwise compromise the confidentiality, integrity, or availability of the account. When compromised, an account must be re-secured in a process we term *account remediation*. In this work, we determine that there are five key phases for account remediation. In order, these are: detecting the compromise, recovering access to the account, limiting access by the attacker, restoring the account state and associated data to the pre-compromise state, and taking action to prevent future compromises.

After having accounts compromised, the authors discovered first-hand how technically

complex and frustrating the task of account remediation can be. We found anecdotally that help documentation provided by web services differs drastically in terms of completeness. When documentation on remediating compromises is lacking, it is much more difficult for users, even technically-savvy users, to remediate a compromise. Therefore, the advice given by web services to help users remediate their accounts is of critical importance. We realized that not only is the advice given to users critical for navigating the process correctly and effectively, but the advice also acts as a proxy for understanding how the organization responsible for creating it views the process.

In this paper, we make the following contributions:

- **Model Account Remediation:** We develop a five-phase model to capture each phase of account remediation, from initial compromise discovery to remediation. We then use this five-phase model to fully represent the range of activities a user may engage in during account remediation in a qualitative codebook.
- **Characterize Webservice Account Remediation Advice:** We use our codebook to evaluate the account remediation advice of 57 popular web services in the United States, providing a window into the resources available to users as well as acting as an implicit measure of web services' own understanding of the issue. We find this advice is sparse and underspecified, especially when we examine activities unique to account remediation. For example, fewer than half of the services studied provide *any* guidance to limit further access by an attacker.
- **Broad Trends and Recommendations:** We find that average phase coverage is higher for services that either are very popular or that have a previously disclosed data breach. We also provide recommendations for web service owners and future researchers.

We note that *account recovery*, defined as the process of restoring a legitimate user's access to an account if credentials are lost or changes, has received substantial research coverage, as we discussed in Chapter 2. However, account recovery is only a single phase of account remediation. Areas such as limiting an account's access and restoring an account's original state are crucial for account remediation, but have received little research attention.

1

¹ Text of this chapter is reprinted with permission from Lorenzo Neil, Elijah Bouma-Sims, Evan Lafontaine, Yasemin Acar, and Bradley Reaves. Investigating Web Service Account Remediation Advice. In the 17th Symposium on Usable Privacy and Security (SOUPS 2021), pages 359-376. USENIX Association, August 2021.

3.2 Methods

In this section, we describe our methods (see Figure 3.1): codebook development (3.2.1), account remediation model creation (3.2.2), ensuring inter-rater reliability among coders (3.2.3), coding account remediation advice from 57 web services (3.2.4), and our analysis of differences in the coverage of account remediation advice among web services based on their popularity and disclosure of data breaches (3.2.5).

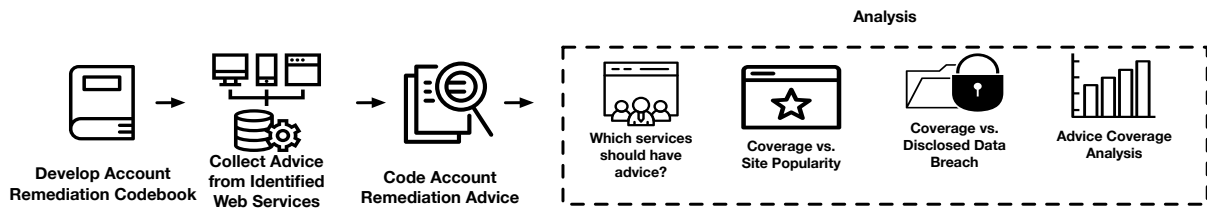


Figure 3.1: Methodology: codebook and model development, data collection and analysis.

3.2.1 Codebook Development

Three authors created the codebook deductively based on nine popular web services' account remediation advice, inductively informed by authors' personal and professional experience with account remediation, and existing research on account recovery, data breach notification and behavior, and authentication. We first annotated nine popular web services' account remediation advice,² then iteratively built and revised our codebook and operationalized the codes. We finalized our codebook when we were able to unambiguously apply it to assess account remediation advice for the initial nine web services. This was evidenced by high agreement when applying the codebook (Krippendorff's Alpha > 0.75 for all three coders for independent coding (Freelon a,b)). In line with recommendations for qualitative coding, we used this score not only to assess our level of agreement, but also to investigate where and how we disagreed (McDonald et al. 2019a; Barbour 2001). If that coefficient was not met when we compared our codes, we used it as an opportunity to better define and disambiguate codes, as well as discuss what causes confusion or disagreement.

² Facebook, Netflix, Skype, Spotify, Twitter, LinkedIn, Google, Yelp, Walmart

The final codebook contains five top-level codes, *compromise discovery*, *account recovery*, *limiting access*, *service restoration*, and *prevention*, which we call the five phases of account remediation, as well as sub-codes that represent concrete advice. For example, in *prevention*, we have a sub-code “enable 2FA”, which describes advice to enable 2-FA for an account to prevent a *future* compromise.

3.2.2 Account Remediation Model

We explain the account remediation process as five phases of account remediation: *compromise discovery*, *account recovery*, *limiting access*, *service restoration*, and *prevention*, corresponding to our codebook’s top-level categories.

Compromise discovery describes a user observing suspicious activity from their account or service that indicates a possible compromise, for example: “If you notice unfamiliar activity on your Google Account, someone else might be using it without your permission.” (Google).

Account recovery describes the process for users to regain access to their account after losing access to it or having it compromised. We differentiate account remediation from account recovery in the sense that account recovery is only one phase in the account remediation process. An example of advice for *account recovery* is: “Change your password or send yourself a password reset email.” (Instagram)

Limiting access describes preventing current and future unauthorized access from adversaries, for example: “Sign out of all devices connected to your account unless you believe your device has been stolen.” (Netflix)

Service restoration describes restoring an account’s original settings, content, or state before a compromise. An example of advice for *service restoration* is: “After signing in, you’ll want to review the recent activity on your account.” (Microsoft)

Prevention describes preventing future compromises by taking steps to further secure an account, like “Never click suspicious links, even if they appear to come from a friend or a company you know” (Facebook).

While advice coverage was not uniform across services, we found that the concept of the top-level categories (our five phases) was present across services. While we theorize that these five phases conceptualize account remediation in general, we do not imply that each specific subcode in each phase has to be covered by all services to provide complete

advice, as service offerings may differ. We established this model to account for a wide range of advice and describe the majority of account remediation steps.

3.2.3 Training and Reliability

Following the development of the codebook, the main author trained two supporting coders on the nine initial web services, again measuring inter-rater reliability to pinpoint and resolve disagreement and to determine the successful conclusion of the training phase. The agreed-upon coding by the three codebook developers was used as ground-truth for training, and once Krippendorff's Alpha consistently exceeded 0.75, we considered the new coders competent to apply the codebook (Freelon b).

After the training phase concluded, the supporting coders then coded the rest of the web services individually. The primary coder independently double-coded a select subset of web services from each supporting coder, usually those that had been subjectively the hardest to code. After each week of independent coding, the primary coder met with each supporting coder separately to resolve disagreements, errors, and confusion, as well as to make sure that coding strategies did not diverge over time.

3.2.4 Collecting Advice from Web Services

In this section, we explain our process for web service selection, how we collect and store the advice, and how we established groups of web services for research questions.

Service Selection Criteria: We referred to two lists generated from the Tranco Website Ranking Service (Pochat et al. 2018) to identify web services of interest. The lists were generated on March 31, 2020 and August 18, 2020. Using these Tranco lists as a reference, we examined web services that were U.S.-based, allowed user online account creation, and provided publicly available account remediation advice. We chose U.S.-based web services since all authors are fluent in English. We excluded adult-content web services from the study, as our research was performed on computers owned by a public university. Finally, we excluded services that were unreachable at the time of data collection.

Finding Advice To ensure the totality of advice collection, we collected account remediation advice from web services by both manually browsing their help pages and through search queries on the website and Google. When navigating the web service, we searched both the help center sections and security settings (if available). We queried the help center

with the template phrases: “My account was compromised” and “My account was hacked”. Once we found a web service’s main page for account remediation advice, we also collected every relevant link mentioned on that page for account remediation. We further extended our collection of advice by Google search querying for any account remediation advice from the target web service based on text snippets we found on advice sites, our own experiences, and anticipating the spectrum of possible user queries. Our Google search queries were the following: “My [web service] account was compromised” and “My [web service] account was hacked”. We added any new account remediation advice that was not found when navigating the web service. This multi-step process ensured that we identified all relevant account remediation advice from a web service. We note that many large companies have separate web services served by the same account management; one example is Google and YouTube. In such cases, we only include an advice policy once.

Content Exclusion Criteria For our analysis, not all information is appropriately considered account remediation advice. For example, we do not consider advice for accounts that were suspended due to actions of the user or suspensions that were self-inflicted. Secondly, we did not include advice within forums or posts by other users on the service or on third-party sites, because such information may be inaccurate, outdated at the time of collection, or from an untrustworthy source. We also exclude advice documents when they consisted *solely and entirely* of a suggestion to contact the service.

We also only collect advice available without requiring a logged-in web service account to replicate the process a user would take if they could not access their compromised account and needed guidance. This strategy also allowed us to collect all relevant advice regardless of the login status. After our initial data collection, we observed that financial services and universities had been almost entirely excluded by this strict criteria. Owing to the importance of these two industries as targets of compromise, we revisited these services to collect publicly available remediation advice. Out of an abundance of caution, in Section 3.3.3 we include results with and without the financial service and university data.

Collected Datasets We divide our collected data into two groups, shown in detailed tables in Appendix A.2. Both groups account for 57 total web services. The *very popular* web services dataset consisted of the top 31 web services (as ranked by Tranco) that were U.S.-based and offered account remediation advice. To this dataset, we added one additional service (Yelp) slightly outside of the Top 31 that had been chosen arbitrarily as a case study during codebook creation. We note that after filtering by our criteria and excluding

combined web properties from the list (e.g., Google and YouTube) our first 31 services span from Google (ranked #1) to Walmart (ranked #184), with our last service (Yelp) ranked 209 at the time of data collection. Therefore, this group consists of 32 web services and we will refer to them as our *very popular* set of web services throughout the paper. We explain in Section 3.2.5 how we define popularity.

Our second dataset, termed the *less popular* web services dataset, consisted of a random selection of 25 services meeting our full criteria with a Tranco rank in the range of 500–1000. Initially, we aimed to collect advice from 32 web services in this range in order to have two equal sets of web services. However, upon coding these web services in the full study, the coders had trouble coding the advice specifically in regards to advice from the phase compromise discovery. The confusion came from the fact that it was hard to differentiate whether advice to discover a compromised account was either solely billing/financial issues or actually other codes related to compromise discovery. Due to this confusion, we decided to discard banking web services in this group of web services, which left us with 25 *less popular* services, as we will refer to them throughout the paper. This specific range was chosen to select a group of web services that were not obscure but was also noticeably different from the very popular web services ranked at the top. Rankings like Tranco in general are rarely linear in correlation with the phenomena measured (or implied).

For example, consider the case of Youtube, Netflix, and Crunchyroll. Youtube and Netflix were ranked 3rd and 9th respectively, while Crunchyroll was ranked 837th. Though Youtube is ranked 3 times higher than Netflix, it is unlikely that YouTube has three times the resources for security than Netflix; nor is it likely the case that Youtube has nearly a three-orders of magnitude larger security budget than Crunchyroll. Consequently, to see if site popularity has an effect on remediation advice coverage, we choose to look at group distances between the rough equivalence classes formed by the broad rank range.

Recording Existence of Account Remediation Advice Using the same selection and exclusion criteria, we analyze all web services that were ranked between 500–1,000 on Tranco (Pochat et al. 2018) for existence of publicly available account remediation advice. We are not coding web services here; we simply check if web services provide public account remediation advice. Therefore, we examine all web services in this range, not just web services with account remediation advice. Once we calculated how many web services fit our selection criteria and provided public account remediation advice, we divided that number by the total number of web services that fit our selection criteria.

We perform this method on two different data sets, each data set however consists of

web services ranked between 500–1,000. Both data sets consisted of web services that were U.S.-based and allowed for account creation. The difference is that the first data set will also include financial or university-based web services that were ranked between 500–1,000. We refer to this data set throughout the paper as the *include financial/university* services group. The second data set is identical but excludes financial or university-based web services ranked between 500–1,000. We define this data set throughout the paper as the *exclude financial/university* services group.

We include two data sets since we cannot confirm if financial-based or university web services provide different account remediation advice to users with a login or belonging to that community. Since our criteria were to only collect advice that was publicly available without a login, we separate our findings for this question. These results will be shown in Section 3.3.3.

Storing Advice When we found all relevant account remediation advice from a web service, we saved PDF versions of the web pages and stored them for analysis. This helped ensure we had a static dataset that did not change as we were coding. This also allowed us to code web services both collectively or individually by analyzing similar PDFs for web service’s account remediation advice.

Coding After the training phase was complete, each new coder coded 22 web services (totaling 44 more web services). Each coder coded the PDF pages from the web service’s advice with Nvivo, in increments of five to nine web services at a time. Once each week, the first author met with both coders separately to go over the overlapping coding results and resolve confusion or disagreements about the coding results. Each coder then corrected their codes or added codes that they missed. Our coding results are in Sections 3.3.1 and 3.3.2.

3.2.5 Differences in Coverage of Account Remediation Advice

In this paper, we seek to understand whether there are significant differences in the coverage of account remediation advice between *very popular* web services and *less popular* web services, and whether there are significant differences in the coverage of account remediation advice between web services with a disclosed data breach and web services without a disclosed data breach. To address these questions, we need to operationalize aspects of these questions, including coverage, popularity, breach history, and group differences. This subsection presents the methods we use for each of these issues.

We operationalize the coverage of account remediation advice as a web service covering all five phases of account remediation in their advice. The range of the coverage of advice is measured from one phase coverage up to five phases coverage. For example, if a web service gives advice that covers only compromise discovery, account recovery, and limiting access, the coverage of the advice for that web service will be a three since it mentioned advice from three phases. We define the coverage of advice for account remediation in this manner because every phase for account remediation is important in successfully remediating a compromised account. However, not every individual code in every phase will be relevant or important for every web service. For example, codes for advice on noticing billing/finance issues will not be relevant for web services that do not handle money transactions or store financial information. Also, web services that do not give users the functionality to install third-party applications will not need advice on how to remove potentially malicious third-party applications. For this reason, if a web service has advice that mentions at least one code from a given phase, that phase will be counted to the coverage of account remediation advice for that web service. While this may overestimate a service's advice (i.e., coverage does not imply a high quality of advice), we can confidently assess services with low coverage and services with high coverage of advice.

We define the popularity of a web service by its ranking on the Tranco Website Ranking Service (Pochat et al. 2018). This ranking service was developed mainly for research purposes and consists of data from many ranking services over a period of 30 days. Tranco lists web services based on their popularity. The top 32 ranked web services we analyze are at the very top of this list, called here the *very popular* group of services. Lower ranked web services on the list such as the 25 randomly sampled web services in the 500-1,000 range are the *less popular* group of services. Our results for comparing the differences in coverage between *very popular* web services and *less popular* web services are shown in Section 3.3.4.

We operationalize “public disclosure of data breaches” by using a well-known database maintained by Troy Hunt on his website “haveibeenpwned” (Hunt 2020) Haveibeenpwned consists of a database of publicly disclosed data breach incidents that have been consolidated and displayed on the website. The database also contains hundreds of database dumps and paste bins containing billions of leaked account credentials. Users then can query this website to search if their credentials such as their emails, usernames, or passwords have been compromised or “pwned.” Users can also check an overview of web services that haveibeenpwned has listed as being breached, and sign up for breach notification. When we define web services to have publicly disclosed a data breach, we refer to

web services that are listed on haveibeenpwned; the *data breach disclosed* group contains 16 web services.

The remaining 41 web services that were not mentioned in the breached list of web services (Hunt 2020) make up our *non-data breach disclosed* group. Our results for comparing the differences in coverage between *data breach disclosed* and *non-data breach disclosed* web services are shown in Section 3.3.5.

In order to statistically evaluate the differences in our two research questions, we perform a Mann-Whitney U Test for both questions. Specifically, we investigate if the means of the distribution of the number of phases within the groups involved in the research questions is significant in difference. Using the Mann-Whitney U scores, we then calculate the magnitude in differences between each group of web service's coverage of advice by calculating their respective effect size (Fritz 2011; Lakens 2013). This effect size is also quantified in Cohen's confidence interval r (Cohen 1988). We follow the interpretations as guidelines provided by Fritz (Fritz 2011), which describe $r = 0.1$ as "small", $r = 0.3$ as "medium", and $r = 0.5$ as "large". The Mann Whitney U Test and other related statistical measures were performed with SPSS software (IBM 2020). We then used these results to calculate the effect size (Fritz 2011).

3.2.6 Limitations

As with any study that involves qualitative coding, this study is subject to the authors' biases, as well as possible differences in coding strategies between coders. We tried to reasonably address these in our investigation by having coders with diverse research backgrounds on our team to allow multiple perspectives to inform the creation of our codebook, and, eventually, the five phase model of account remediation. We also diligently refined our codebook and the codes' explanations in order to allow independent coders to arrive at similar assessments, and regularly controlled for divergent strategies, discussed differences and resolved disagreements.

Additionally, due to the nature of our study, we cannot provide ground truth about the differences in the coverage of account remediation advice between different groups of web services. Our definition in the coverage of advice does not take into account the length or depth of the advice, rather a metric for how many phases in account remediation it covers. We also do not provide ground truth for the applicability of all of the codes in our codebook to web services. Most of the codes in our codebook represent advice that can be broadly

applied to all web services. However, some codes that we developed during our codebook development like “observe billing” or “finance issues” or “observe a third party account connected” do not apply to all web services. Therefore, we explain the results for specific codes like this with the caveat that they may not be broadly applicable to all web services.

We only collect advice from web services when it was publicly available without an account login. Some web services may provide additional account remediation advice once a user is logged in. We collected advice in this manner to replicate the process of finding account remediation advice, in the case where the account owner cannot access their account. For our coding results in Sections 3.3.1, 3.3.2, 3.3.4, and 3.3.5, we only include web services that provide publicly available account remediation. In Section 3.3.3, we include two versions of results in which we exclude web services that may provide additional account remediation advice given an account login.

Lastly, we only included U.S.-based web services in this study. We wanted to ensure that all coders could fully interpret and code the web services we selected for this work. Since the only language that every author could fluently speak is English, we limited ourselves to U.S.-based web services.

3.3 Account Remediation Advice Coverage Results

In this section, we discuss the results of our codes and implications behind the results. In Section 3.3.1, we provide the overall coverage of the phases for account remediation advice from the web services. In Section 3.3.2, we look at each phase individually and examine the coverage of their respective codes within the web services. In Section 3.3.3, we report how many of bottom 500 ranked web services provided users with publicly available account remediation advice. We present this report with the inclusion of financial web services and university web services and also without financial web services and university web services. In Section 3.3.4, we present our results for investigating the differences in the coverage of account remediation advice between *very popular* and *less popular* web services. Similarly in Section 3.3.5, we present our results for investigating the differences in the coverage of account remediation advice between *data breach disclosed* services and *non-data breach disclosed* services.

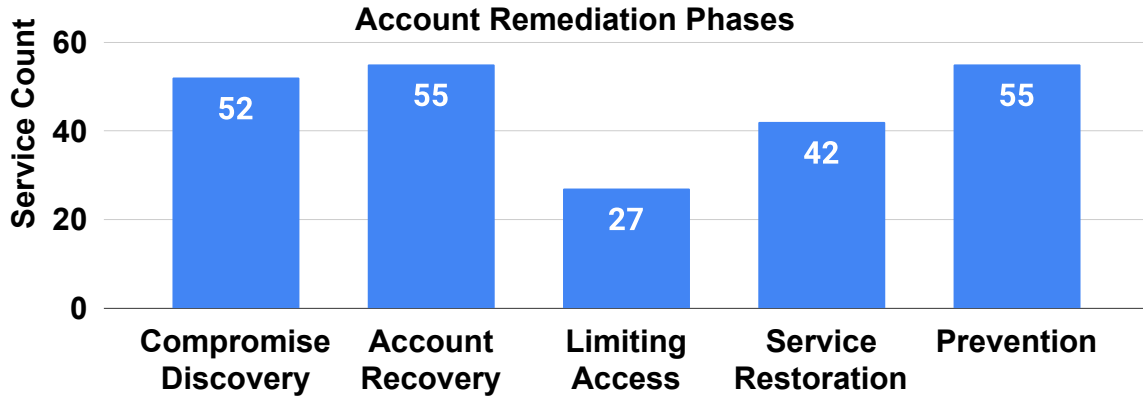


Figure 3.2: Bar graph of all account remediation phases among web services. Limiting Access advice is mentioned in less than half of the web services we analyzed. Service Restoration advice was mentioned in 74% of the web services. All other phases were mentioned by at least 90% of the web services we analyzed.

3.3.1 Overall Phase Coverage

Sections 3.3.1 and 3.3.2 reflect results from coding all 57 web services. Advice for compromise discovery, account recovery, and prevention was mentioned by 91%, 96%, and 96% of all web services, respectively. These were the only phases that were covered in at least 80% of account remediation advice from web services. On the other hand, advice for limiting access was mentioned by 46% of web services and advice for service restoration was mentioned by 75% of web services. Figure 3.2 represents web service counts for every phase in the account remediation model. The service count in the graph indicates how many web services mentioned at least one code from a specific phase.

The phases of compromise discovery, account recovery, and prevention are not only widely addressed by most web services, but also represent areas that have been heavily researched by the security community. These three phases, however, do not fully cover the process of account remediation. Limiting the access of an account and restoring an account’s original settings are fundamental for account remediation. Without it, the account remediation process is not complete, and a compromised account may still remain vulnerable. Still, more than half of the web services we investigated did not mention any advice for limiting an attacker’s access.

Out of the total 57 web services we analyzed for account remediation advice, only 39% managed to mention advice from all five phases. 74% of web services mentioned at least

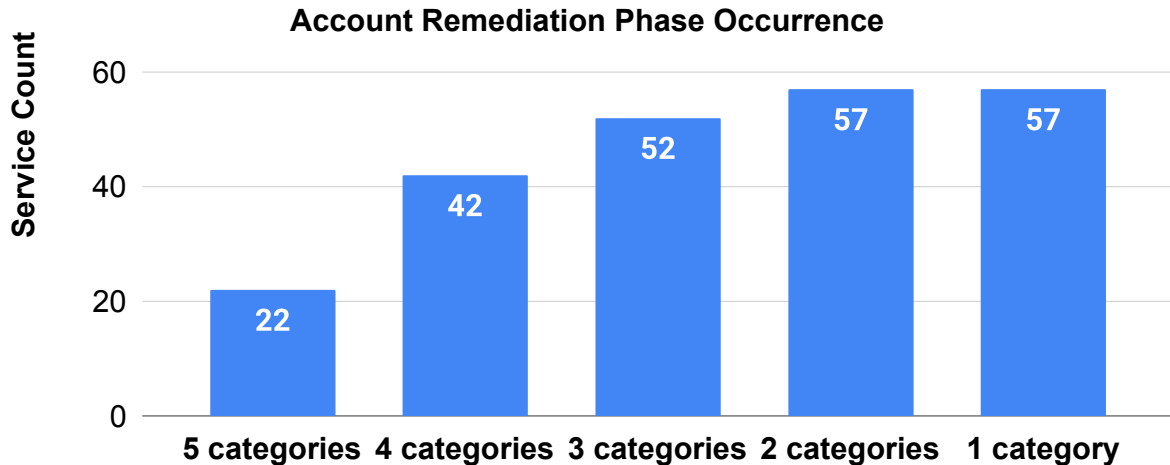


Figure 3.3: How many web services mentioned at least n amount of web services, where n is either at least 5,4,3,2,or 1 phase. Only 39% web services gave advice for all five phases.

four account remediation phases. 91% of web services mentioned at least three account remediation phases. Lastly, all 57 web services mentioned at least two account remediation phases. Figure 3.3 shows these results from coding all 57 web services.

The consequences of these results require careful consideration. On the one hand, our results for security advice most unique to account remediation (limiting access and service restoration) would seem to indicate that web services are neglecting these two phases. On the other hand, while we believe our model is sufficiently general to capture the account remediation process, there may be cases where it is not necessary to cover all five phases explicitly. Consider a hypothetical service that recommends completing the account recovery process, and it happens to log out all logged-in sessions. The service’s advice may not reflect any limiting access content because it is automatically handled. Without ground-truth knowledge about each web service’s internals, it is difficult to determine which case applies to a particular web service. Taken together, it is clear that future work should determine if remediation phase coverage is low because it is neglected or if it is simply not necessary.

3.3.2 Content Analysis by Phase

Compromise discovery: Compromise Discovery involves observing activity from an account or service that indicates a possible compromise. Our results for the compromise

discovery codes are shown in Figure 3.4. *Only 11% of the codes in this phase were covered by at least half of the web services.*

Advice for discovering unauthorized or suspicious activity was recorded in 68% of the web services. This was the only advice in compromise discovery however that was mentioned in at least half of the web services. A possible reason for this could be that all of the advice in this phase can be related to unauthorized or suspicious activity, and the code itself is much less specific compared to other codes in this phase. This is a broad interpretation of compromise discovery since there are multiple methods of compromise discovery.

Advice to discover an email change or password change was mentioned in 12%, and 21% of web services, respectively. The majority, if not all, of web services with account creation store a user's email address and password and allow users to change them as well. Observing that either of these identifiers changed within an account is a strong indication of a possible compromise. Still, even the union of the coverage of advice for discovering a changed password and changed email address reached no more than 33% of web services we investigated. This is a clear oversight of advice coverage on the part of web services.

Advice noticing an explicit notification and observing unauthorized logins was mentioned in 30% and 35% of the web services we investigated, respectively. We wanted to code advice for users discovering account compromises from explicit notifications from the service, or by observing unauthorized logins on their accounts. From this, we also concluded that users could observe unauthorized logins due to an explicit service notification, or by examining their account as well. Therefore, we created a code for noticing explicit service notifications about a compromise and a code for observing unauthorized logins that includes coverage from the explicit service notification code, while not being exclusive to it. With these results, we present the caveat that we do not confirm if all web services give users the functionality to observe log-ins on their accounts. Therefore, the results for our code "observe an unauthorized login" may not be broadly applied to all web services.

Advice to discover a social media/third party account connected and billing/finance issues were mentioned in only 5% and 35% of web services, respectively. While these results do reflect low coverage across web services, we can not confirm how many web services in our study implement billing or finances into their functionality for users. We also can not confirm if all web services in our study allow users to connect a social media or third-party account to their main account.

We look to our results in coding limiting access advice later in this section and compare the results of the code "Remove third party access." This specific code, "Remove third party

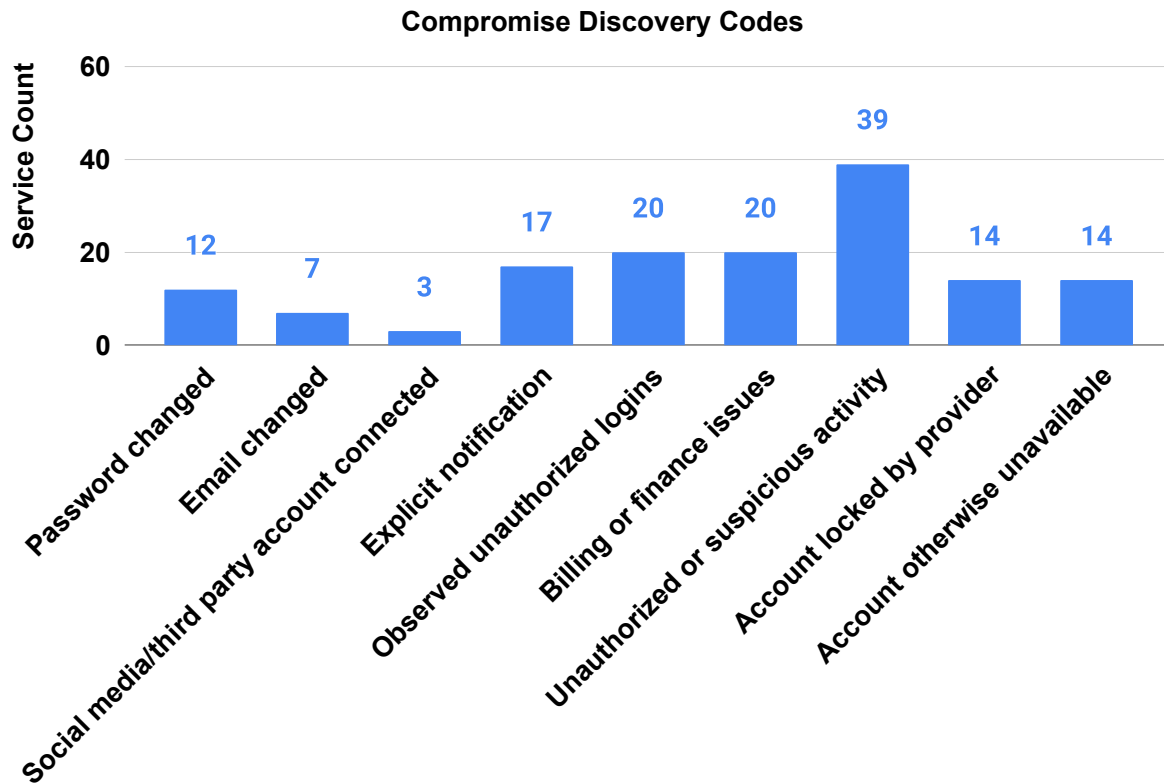


Figure 3.4: Bar graph of Compromise Discovery codes among web services. Unauthorized or suspicious activity was the highest covered code with 39 web services. No other code was mentioned in more than half of the web services.

access”, was mentioned in 18% of web services. The difference in coverage between this code and our code in this category, “social media/third party account connected,” shows that at least 12% of web services that allow users to connect a social media or third account are not advising users to notice a new social media or third-party account when discovering a compromise.

Overall, compromise discovery advice was sparsely covered. Only one code in this phase was covered by at least half of the web services. Most of the codes in this phase can either be broadly applied or covered at a higher usage given other results we recorded in other phases. Most of the advice in this phase is also cheap in implementation but important to discovering a compromised account. Web services have much room for improvement in their coverage of compromise discovery advice.

Account recovery: Account recovery provides a means for users to recover their account after losing access to it or having it compromised. Our results for coding this phase are shown in Figure 3.5. *66% of the codes from this phase were covered in at least half of the web services.* This phase is highly covered by web services and continues to be prioritized, even as a means to remediate compromised accounts.

Advice to initiate a password reset or to change a password was covered in 91% of web services. This advice was also the highest covered code out of all phases in this study. It was the most common method for advising users to recover their compromised accounts.

Advice to advise users to engage in customer service to recover a compromised account was covered by 63% of web services. Some services require contacting customer service for account recovery processes. Customer service for account recovery involves assisting users in recovering a compromised account with a guided process or interaction with a service client. This is different from other customer service processes that services may offer outside of account recovery. While we recognize this advice was not covered universally among web services, it may not be reasonable to have users go through customer service every time to recover their account or reset their password. However, keeping customer service as an optional route may be more beneficial to users.

Advice to reset passwords and to engage in customer service to recover an account were both covered in over half of the web services. These results can imply that not only is account recovery prioritized in account remediation advice, but mainly in the forms of password reset advice and customer service support.

We observed advice for running endpoint security to recover an account was only covered in 14% of web services. The low service count could be the result of authors of account remediation advice not considering endpoint security. Also, correctly running anti-virus software is highly technical and possibly beyond the reach of most users. It might be unclear to the extent of how much antivirus or other harm remediation measures help remediate online account compromise. This can imply that web services may not view endpoint security options as a viable solution or prioritize it for account recovery purposes.

Limiting access: Limiting account access is defined as preventing current and future unauthorized access by adversaries. *Limiting Access advice was the lowest covered phase in the study, reaching only 47% of web services.* Less than half of the web services in our study advised users to manage the access of their account, and thus not prioritizing an important step in account remediation. Advice for limiting an account's access includes signing out of instances of an account, reviewing active sessions, and removing access from third-party

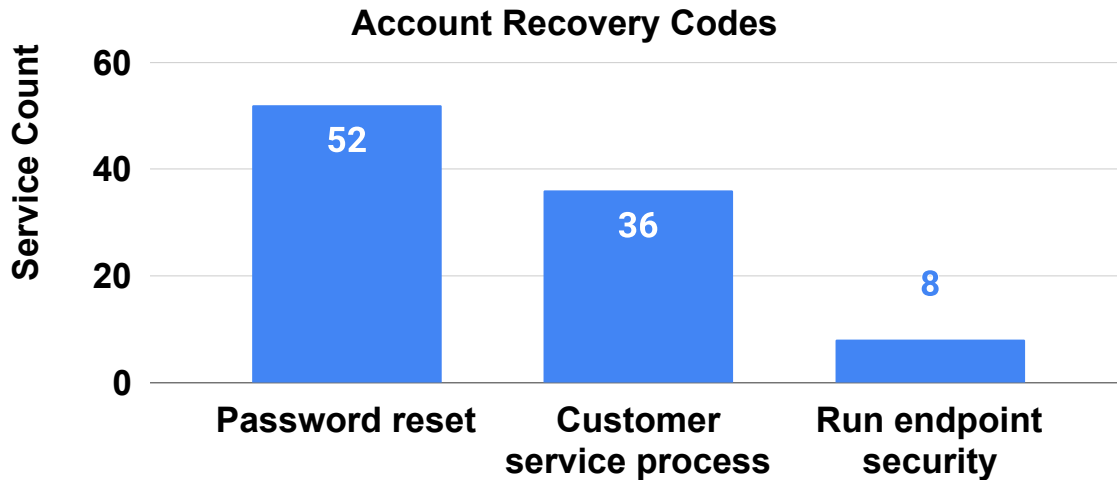


Figure 3.5: Bar graph of Account Recovery codes among web services. Password reset was mentioned by 91% of services and customer service support was mentioned by 63% of services.

applications. The results for coding this phase are shown in Figure 3.6.

Advice for signing out of an individual instance or all instances of an account were covered by only 26% and 14% of web services, respectively. All services allow users to sign out of an account and many allow to sign out of multiple account instances, yet the union of these two codes was only covered by 40% of web services. This coverage is insufficient given that all web services allow users to sign out of an instance or multiple instances of their account and it is an important step in managing the access of an account.

Advice for reviewing active sessions also was represented with a code that was only present in 12% of web services. We also record this finding with the caveat that we lack ground truth for how many web services provide users the ability to check for active sessions of their account. However, we explain in Section 3.4, why we recommend this functionality be implemented in web services and then provided in account remediation advice.

Advice for removing third party access was only present in 18% of web services we investigated. This is important to note since advice for discovering a new social media or third-party account connected to an account in the compromise discovery category was only mentioned in 5% of web services. All of the advice in this phase is underwhelmingly covered given its importance to secure the access of a compromised account.

Service restoration: Service Restoration advice involves restoring an account's original

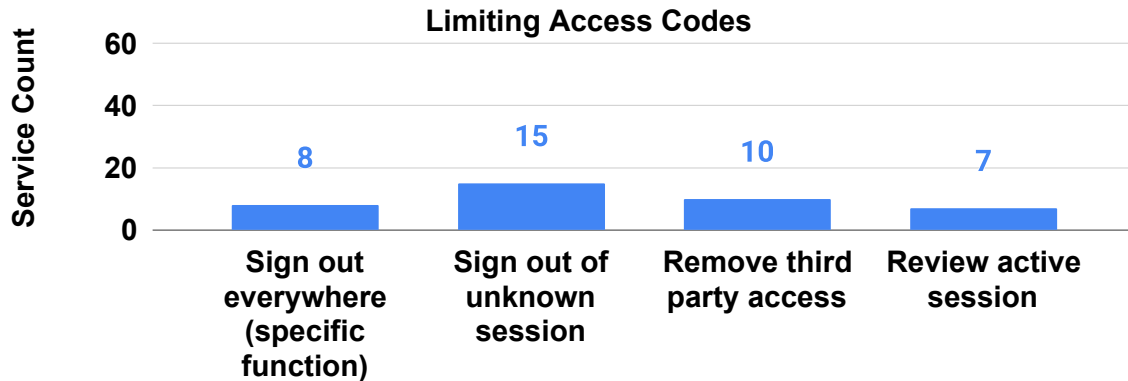


Figure 3.6: Bar graph of Limiting Access codes among web services. No single code was mentioned in more than a third of the web services.

settings or information to how it was before the compromise. *74% of web services mentioned advice for service restoration, yet none of the specific codes in service restoration were covered by at least half of the services.* The results for coding this phase are shown in Figure 3.7. Advice from this phase is also insufficient in coverage among web services.

Advice for verifying user information, verifying account settings, and reviewing and/or removing activities or content were each recorded in 42%, 28%, and 39% of web services, respectively. These are extremely low percentages for advice that should apply to most, if not all, of the web services we analyzed. All web services in this study store information about the user, settings for the user, and activity by the user. Therefore, there should be advice to verify all of this information. Yet, none of the codes that represent this advice are mentioned beyond 42% of web services investigated.

Lastly, advice to seek customer service support in this phase received a low percentage: 23% of web services. This percentage differs significantly in coverage than the service count for customer service support for account recovery which was mentioned in 63% of web services. This could imply that most services are more likely to prioritize customer service support advice for account recovery, or they do not prioritize customer service for service restoration purposes.

Prevention: Prevention is defined as taking further steps to further secure an account. *Out of the total 11 codes in Prevention, four were represented in at least 60% of the web services investigated.* This category also held six of the top ten most covered codes in the codebook (strong password advice, secure email advice, enable 2FA, check/modify related

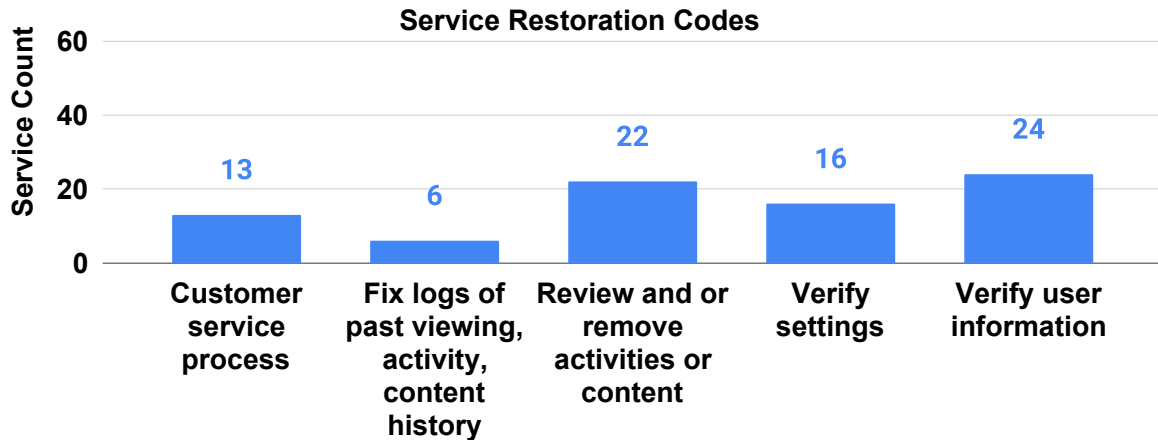


Figure 3.7: Bar graph of Service Restoration codes among web services. No single code was mentioned in more than 42% of web services.

accounts, enable endpoint security options, and keep software updated). Results for coding this phase are presented in Figure 3.8.

Advice to maintain strong passwords was the highest mentioned code in this category with 88% coverage. This was the second individual highest covered code right behind the advice to initiate a password reset to recover an account (91% coverage). This means that advice for password security amounted to the two highest codes and therefore the highest coverage out of any advice for account remediation. This could be a result of the vast industry and academic work on password security. It could also mean that web services believe strong password advice is very crucial to account remediation.

Advice on securing emails, enabling two-Factor Authentication, and checking or modifying related accounts was covered in 72%, 70%, and 61% of web services, respectively. Similar to strong password advice, secure email advice and two-factor authentication advice also represent areas that are heavily researched by the research community and are popular among web services.

Running endpoint security options and keeping software up to date advice were both mentioned in 47% of web services investigated. Interestingly, the coverage in this phase for running endpoint security was significantly higher than advice for running endpoint security for account recovery (14%). This shows authors of advice for account remediation were more likely to advise users to run endpoint security options to prevent an account compromise instead of recovering an account from compromise. However, given that

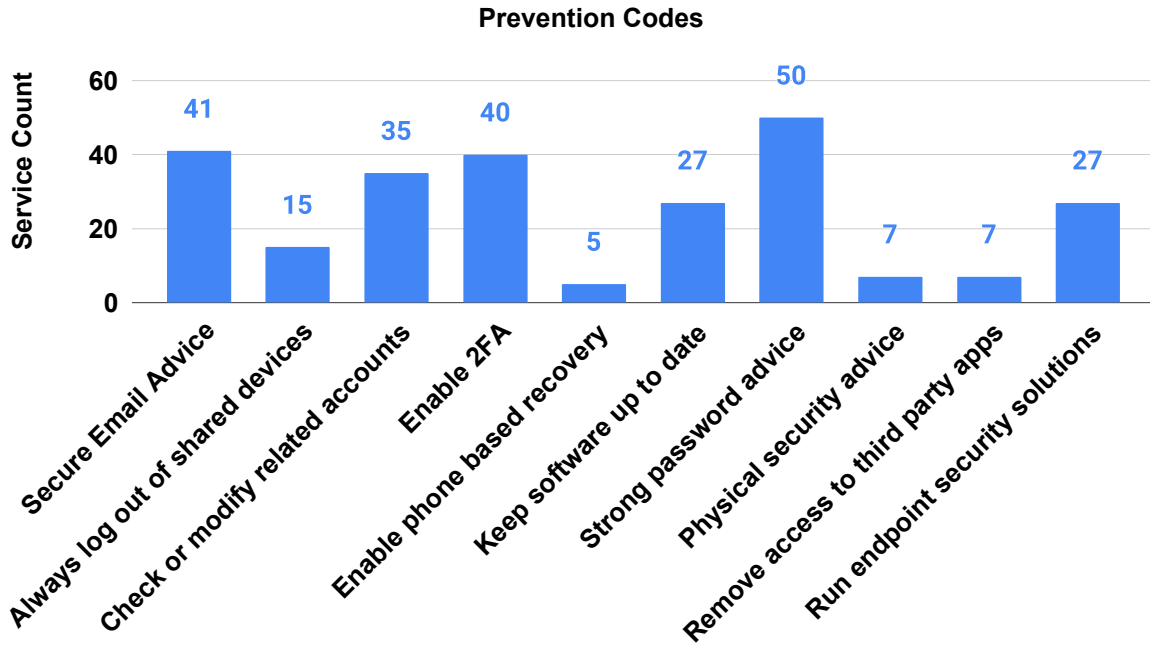


Figure 3.8: Bar graph of Prevention codes among web services. Four out of 11 codes were mentioned in at least 60% of web services and strong password advice was mentioned in 88% of web services.

it is unclear how effective running endpoint security options are towards recovering a compromised account, it is also unclear as to how effective it is in preventing a future compromise.

Notably, prevention advice generally focused on shifting responsibility to other services or the user. While not explicitly coded for, very few services discussed reporting breaches or security flaws in their own service. For example, Netflix states that "If [users] believe [they've] found a security vulnerability on a Netflix property or app, we strongly encourage [them] to inform [Netflix] as quickly as possible and to not disclose the vulnerability publicly until it is fixed." In the worst case, Fandom.com prefaces its prevention advice with the statement that "there is a possibility that if your account is hacked you will need to create a new account" and implies that security is solely the responsibility of the user.

3.3.3 85% of Web Services did not provide Account Remediation Advice

In our *include financial/university* web service data set, 220 web services allowed users to create public accounts and were U.S.-based. *Of these 220 web services, only 15% of these web services gave publicly available account remediation advice.* In our *exclude financial/university* web service data set, 195 web services allowed users to create a public accounts and were U.S.-based. *Of these 195 web services, only 12% of these web services gave publicly available account remediation advice.*

The majority of web services in our study that were U.S.-based and allowed for user account creation did not provide users with public advice for account remediation. This is alarming since we made sure to only collect account remediation advice from a web service if the advice was publicly available and did not require users to log in. A user with a compromised online account needs to have access to such advice even if they cannot access their account. If this advice is not made publicly available, let alone created at all, then users are left with significantly less help in successfully remediating their compromised accounts.

3.3.4 Coverage of Advice versus Popularity

In this section, we give our results from investigating the differences in the coverage of account remediation advice between *very popular* web services and *less popular* web services. As stated in Section 3.2.5, we define the coverage of account remediation advice as the number of account remediation phases that are discussed by a web service.

Our objective is to see if there are differences in the number of phases covered within account remediation advice for web services of vastly different popularity. We performed a Mann-Whitney U Test in which we define the following null hypothesis: the distribution of the number of phases mentioned in account remediation advice is similar across *very popular* web services and *less popular* web services. We perform this test to discover if the number of phases between the two groups of web services is significant in difference.

The mean number of phases mentioned by *very popular* web services was 4.3 with a standard deviation of 0.90. While the mean number of phases mentioned by *less popular* web services was 3.6 with a standard deviation of 0.90. Using a Mann-Whitney U test, we find a statistically significant difference in the mean number of phases covered by the two groups ($U = 224$, $z = -2.994$, $p = 0.003$). Using these test scores, we calculate an effect size

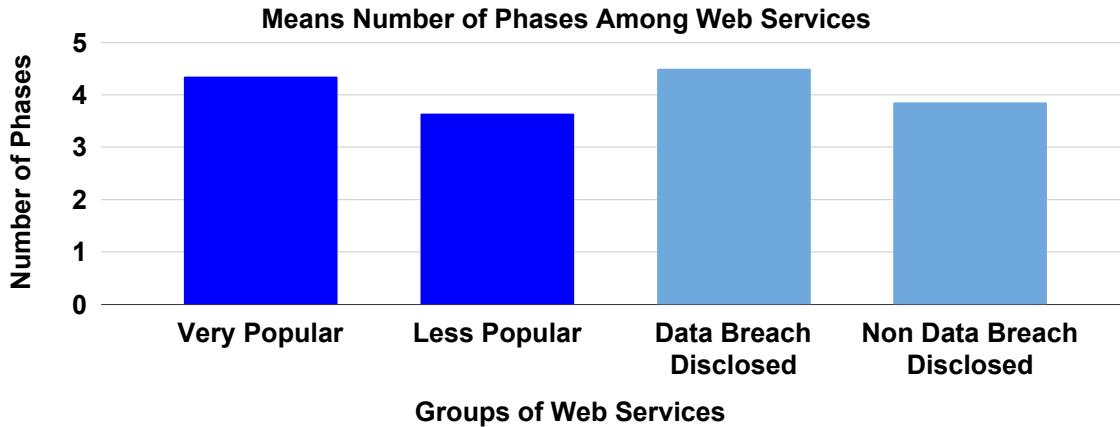


Figure 3.9: Graph of the mean number of phases covered in account remediation by all experimental groups. *Very popular* web services had a higher mean count of phases mentioned in their account remediation advice than *less popular* web services. *Data breach disclosed* services had a higher mean count of phases mentioned in their account remediation advice than *non-data breach disclosed* web services.

$r = 0.397$, which is considered to be a “medium” effect size (Fritz 2011; Lakens 2013).

It is plausible that *very popular* web services have more incentive to provide users with account remediation advice since they have more users creating accounts than less popular web services. Not only would they have more users, but there may also be a higher importance or usage of accounts with very popular web services.

However, there are important web services that are not *very popular*, but are likely to also provide extensive account remediation advice. Financial and banking web services are also important to users, and compromised accounts from these web services can impact a user’s finances or potentially compromise their identity. Many banks provide both advice for account remediation and identity theft and also give users resources to contact for further assistance.

3.3.5 Coverage of Advice versus Disclosed Data Breach

In this section, we show the differences in the coverage of account remediation advice between *data breach disclosed* web services and *non-data breach disclosed* web services.

Our objective is to see if there are differences in the number of phases covered within account remediation advice for web services that have or have not publicly disclosed a data breach. We performed a Mann-Whitney U Test in which we define the following null hypothesis: the distribution of the number of phases mentioned in account remediation advice is similar across *data breach disclosed* web services and *non-data breach disclosed* web services. We perform this test to discover if the number of phases between the two groups of web services is significant in difference.

The mean number of phases mentioned by *data breach disclosed* web services was 4.5 with a standard deviation of 0.63. While the mean number of phases mentioned by *non-data breach disclosed* web services was 3.8 with a standard deviation of 1.0. Using a Mann-Whitney U test, we find a statistically significant difference in the mean number of phases covered by the two groups ($U = 210$, $z = -2.217$, $p = 0.027$). Using these test scores, we calculate an effect size $r = 0.294$, which is considered to be approximately a “medium” effect size (Fritz 2011; Lakens 2013).

These findings may suggest that *data breach disclosed* web services have updated their account remediation advice once their compromised data was publicly known. The breach may have influenced a service to improve their systems and the resources they provide to users to secure their accounts. Interestingly, despite having the experience of a data breach, none of the web services which had disclosed a breach on haveibeenpwned explicitly mention reporting security flaws in the service to mitigate or prevent breaches.

Finally, we note that the analyses of differences of advice based on popularity and history of disclosing data breaches are preliminary and correlational. More work would be needed to confirm a causal relationship between a web service’s coverage of account remediation advice and its popularity or history of disclosing data breaches.

3.4 Discussion

In this section, we discuss recommendations for implementing account remediation advice for web services. We also discuss what future work can be done to further this investigation.

Account Remediation Model: While remediation for each web service may have domain-specific concerns like fixing a playlist or recovering documents in cloud storage, our validated codebook provides evidence that the majority of account remediation steps are general, if not universal. Each phase in our codebook was constructed by analyzing multiple

popular web services and creating codes that be broadly applied. We note that if one defines account remediation as “reversing the consequences of compromise,” one must have all five phases for successful account remediation. One cannot claim an account is remediated until the compromise is discovered, user access is regained, the attacker has lost access, the account is restored to its pre-compromise state, and re-compromise is prevented. If any step is neglected, either a compromise is not remediated or the account will simply be re-compromised.

Our codebook also provides flexibility for domain-specific concerns as well. As discussed in Chapter 2, specific phases of account remediation such as discovering compromised accounts (Egele et al. 2015; Ruan et al. 2015; Cidon et al. 2019; Karimi et al. 2018; Shay et al. 2014), recovering compromised accounts (Parkin et al. 2015; Schechter et al. 2009; Bonneau et al. 2015, 2012; Huh et al. 2017), and preventing compromises through general security practices (Mayer et al. 2021; Redmiles et al. 2020) have been researched and implemented. However, we are the first to conceptually define account remediation into a five-phase structured process. While the variations between services mean that account remediation advice cannot be totally centralized, we believe our codebook could be used for consumer advocates (such as the FTC) as the basis of public information campaigns and guides to help users in the complex task of account remediation.

On Service Responsibility: As mentioned in Section 3.3.2, much of the remediation advice given by services focus exclusively on account compromise resulting from other services or user error. They suggest that compromises may result from poor password choice, password reuse, falling victim to phishing, compromise of a “master” account like an email account, or malware infection. An example of advice following this tone is the following: “Don’t worry, we have no indication that the Walmart systems have ever been compromised, but there are steps you should take to protect your personal information if you suspect unauthorized access or a phishing attempt”. Services very rarely mention the possibility of a security flaw in their own service, even when they have previously disclosed a breach. While it may be the case that the source of most compromises is from external sources, companies should not completely shift responsibility onto individuals.

Additionally, in some cases, users are limited in their ability to remediate an account. For example, banks do not allow users to unilaterally revoke a transaction after completion, and many web services automatically lock accounts based on indicators of compromise. An argument could be made that if web services have the best visibility and ability to detect compromise, they should also be able to assist users proactively, if not automatically, in

remediating the effects of that compromise. On the other hand, if it is true that account compromises mostly originate from external security problems, it would be unfair to put this burden solely on the web service. Similarly, the web service may have an incomplete perspective on what actions around the time of an account compromise were authorized or not. By analogy, credit card companies have regular monitoring for anomalous transactions, and in many cases can automatically block fraudulent transactions, even when caused by an external breach. Still, credit card companies often have to contact their users to confirm or deny specific anomalous charges. We recommend that web services consider to what extent they can automate remediating compromised accounts in order to balance responsibility with best serving users. We also suggest that language should be added to account remediation advice to encourage users to report security flaws with the service rather than focusing only on external causes of hacking.

Another question is what role, if any, law enforcement agencies have to play in identifying and prosecuting account compromises (especially in the furtherance of other criminal activities). We noted that 15 web services mention some form of evidence gathering of an account compromise alongside account remediation advice. However, we also note that computer crime is notoriously difficult to bring to prosecution, so it is arguable to what extent this would be helpful to current or even future victims.

Recommendations: Web services should, as a best practice, provide a mechanism to review account activity, including logins and actions that change the state of an account (purchases, password or preference changes, settings, user information.) Services should also provide better guidance on what “unusual activity” means through specific examples such as changed passwords, changed usernames, or changed emails. Owing to the large amount of prior work on account recovery, we recommend readers see the recommendations of prior work (Parkin et al. 2015; Bonneau et al. 2015; Schechter et al. 2009; Huh et al. 2017). All web services should also provide an interface to show all active log-in sessions and/or access permissions. This interface should also allow a user to revoke access for any or all current sessions. Along with the recommendation to show account activity, there should be an interface allowing users to revert changes made to their settings or remove unauthorized content. While not specifically coded for in our study, we observed that only six services provided a method to restore content deleted in an account compromise.

Enforcing mandatory customer service for account remediation purposes will inform the web service directly while also potentially discovering a large scale data breach. On the other hand, it potentially increases the effort on part of both the user and web service. Also,

if mandatory customer service is not staffed 24/7, there may be consequential delays in preventing further damage from the compromise. This is why optional customer service may be a better feature to have, especially for complex remediation cases, because users without significant technical understanding of the compromise may need additional support. Finally, we observed a high variance in the prevention advice given by web services for what is largely the same problem, implying that many individual web services have incomplete prevention guidance. Similar to the work done by Redmiles et al. (Redmiles et al. 2016b, 2020), there is an abundance of general prevention advice but a lack of advice prioritization.

Future Work: Future work should explore more usable or contextual guidance. Some of the steps in account remediation are technically complex to perform for users. Making the process of account remediation more usable and easier to follow will better aid users in remediating their accounts. For example, Facebook actually implements a chatbot-style wizard for guiding users through account remediation. It consists of easy to read diagrams that prompts users if they recognize information or settings on their account that is presented to them by the chatbot wizards. Future work could evaluate these approaches and explore ways of generalizing this approach to be usable for other types of web services beyond social media. Additionally, it is worth exploring to what extent a service could certify that an account has been remediated, or what assurances could be provided to users that their accounts have become “safe.”

3.5 Conclusion

Online account compromises have become rampant, and anyone with an online account is susceptible to having their account compromised. The resources that help users remediate a compromised account should cover all the necessary procedures to help users re-secure their accounts. We investigated publicly available advice for account remediation from both top-ranked web services and lower-ranked web services. We identified important phases for account remediation that are not only sparse in coverage but also are not addressed by a significant amount of popular web services that provide account remediation advice. Also, the amount of web services we studied that even provide users with publicly available account remediation advice is critically low and did not surpass at least 15% of the total web services we analyzed that allow users to create accounts. Lastly, we discovered that highly

ranked web services and web services with a previously disclosed data breach presented more complete coverage of their account remediation advice than other web services. Our analysis of the coverage of account remediation advice presented important areas that are lacking in attention, to which we explain credible recommendations to both bolster the advice and the process of account remediation.

CHAPTER

4

WHO COMES UP WITH THIS STUFF? INTERVIEWING AUTHORS TO UNDERSTAND HOW THEY PRODUCE SECURITY ADVICE

4.1 Introduction

Most users of technology receive advice from experts to keep themselves and their devices safe. The overarching goal of security advice is to provide reliable and up-to-date security awareness and recommendations to end users so that they can practice secure behaviors. Employees of organizations may receive regular security advice and training from their employers, students may receive security advice from their schools and/or universities, and multiple government organizations including the US Department of State offer security advice to the general public (of State 2022). “Second-hand” security advice abounds, pro-

liferated by media outlets (Fulton et al. 2019b; Nicholson et al. 2019), websites (Redmiles et al. 2016a; Herley 2013; Turner et al. 2021; Redmiles et al. 2020), and peer users (Pfeffer et al. 2022; Rader et al. 2012).

End users are thus exposed to a sea of security advice and are unsure which advice is best suitable for them (Rader and Wash 2015; Redmiles et al. 2017, 2020). Experts also struggle to agree on which security advice should be prioritized. Previous related work demonstrated that experts list a total of 118 studied security behaviors as being the “Top 5” things users should do to protect themselves online (Redmiles et al. 2020). A lack of consensus on the most important security imperatives leaves end users to themselves to prioritize and implement security advice. Authors who write security advice thus have to decide which advice is most important for their target audience, who already struggle to prioritize security advice.

In this paper, we seek to identify ground truth and root causes for why security advice varies in quality and prioritization by going to the source: authors themselves. We report findings from a semi-structured interview study with 21 authors of general security advice where we discussed the full process of advice creation from beginning to end. By “general security advice“, we mean security advice just for the general public, or end users with a “general public level knowledge” of security. We investigated authors’ backgrounds and motivations, asked about individual and organizational processes surrounding advice, and asked what they felt were the most challenging aspects of advice creation. In reporting the outcomes of these interviews, we present the following key findings:

Content Creation Authors overwhelmingly perceive setting advice scope and technical level as a central challenge. Decisions about scope have major consequences on every other aspect of advice creation. Advice writers also report revising content when novel security threats or events prompt ad hoc additions. These challenges partially explain the overproduction and undercuration of security advice (Redmiles et al. 2020).

Internal and External Influences Authors reported input or oversight from a wide variety of stakeholders in their organizations, including technical and operational staff, legal departments, and even C-level executives. Legal regulations and technical standards also heavily influence advice content, with some organizations seeking comprehensive compliance or congruence with multiple sources. Authors consult a wide array of authoritative sources, with little consistency from author to author.

Recommendations Section 4.4.1 discusses implications of our findings and provides future recommendations. These recommendations include research on sound methodologies for curating and prioritizing advice, research establishing guidance on advice communication for users with varying levels of technical expertise, and for authors to proactively plan advice reviews to improve focus and not just augment advice.

1

4.2 Methods

We conducted 21 semi-structured interviews with authors of online security advice between September 2021 and March 2022 to understand the processes and decision-making that go into writing general security advice. We obtained written transcripts of audio interview recordings and analyzed the transcripts through deductive and inductive coding. Written transcripts were de-identified by replacing personally identifiable information (participant names, organizational names) with pseudonyms such as F001 for freelance workers, I002 for industry workers, and U007 for university IT and security workers. Participants were informed of the research goals and how their information would be protected in our screening survey consent forms. Our study protocol was approved by our University’s Institutional Review Board (IRB).

4.2.1 Participant Recruitment

This project focuses on a specific expert population: authors of general security advice. We define such authors as those with professional experience in drafting content for general security advice. We recruit participants through purposive sampling of those who qualify through various recruitment channels, namely personal and professional contacts, social media advertising, recruitment on the freelancer platform Upwork, and directly emailing those who manage university security advice websites. We first directly recruited qualifying personal and professional contacts, then we posted messages on professional social media (Twitter, LinkedIn, and industry mailing lists) to solicit potential participants. We also advertised our study on the popular freelancer website Upwork (Upwork 2022) to

¹ Text of this chapter is reprinted with permission from Lorenzo Neil, Harshini Sri Ramulu, Yasemin Acar, and Bradley Reaves. Who comes up with this stuff? Interviews with security advice authors. In the 19th Symposium on Usable Privacy and Security (SOUPS 2023), pages 283-299. USENIX Association, August 2023.

recruit freelancers with professional experience in writing general security advice. After every interview, we asked participants if they knew other individuals that might qualify for our interview study. Finally, we reached out to IT help desks and information security or technology departments from U.S. universities found through a top national universities rankings website (News 2022). Here, we contacted 109 universities that provided both general security advice on their website and an email contact to either their IT help desk, information technology department, or security department. We contacted all potential participants with a recruitment email, linking to our public website which presented a study overview, supplementary information, and a link to the screening survey consent form.

Once we identified a potential participant and they replied with interest, we sent them a screening survey and informed consent form through Qualtrics (Qualtrics 2022). The screening survey described our research goals at a high level, asked for consent to be video and/or audio recorded for the interview, and requested basic demographic information from the participant (Qualtrics 2022). The survey also acted as a qualifier to ensure that the participant had prior professional experience with writing general security advice. Our screening survey asked participants to report on their security experience, such as how long they had been writing security advice, for what companies, and how they learned to write advice. Once eligible participants filled out the screening survey, we scheduled a one-hour interview with them. Participants were compensated with \$30 per half hour for their participation in the interviews.

We concluded recruitment when we reached theoretical saturation; i.e., we discovered that participant responses were not presenting new information beyond data we had already collected (Saunders et al. 2018). Of the 21 participants, 12 were freelance workers, 6 were university security department staff, and 3 were industry workers. 9 of the freelancer workers wrote general security advice for external organizations, similar to a consulting role. 3 of the freelancer workers and 1 of the industry workers wrote general security advice for entities within their own organization or subsidiary organizations. The remaining 2 industry workers and 6 university employees wrote general security advice for the public consumer associated with their networks. Participants of different roles held different levels of involvement for specifically prioritizing the advice content. Awareness experts are communication specialists who reported “translating” advice from security employees to the general public, where security experts, technical experts, and analysts reported researching, brainstorming, or reviewing the audience’s environment to formulate ideas for

Table 4.1: Participant Demographics.

Gender	
Men: 13 (61.9%)	Women: 8 (38.1%)
Target Audience	
University Members: 6 (28.6%)	External Consulting: 9 (42.9%)
Public Consumers: 2 (9.5%)	Internal Consulting: 4 (19.0%)
Advice Generation Role	
Analyst: 3 (14.3%)	Security Expert: 14 (66.7%)
Awareness Expert: 3 (14.3%)	Technical Expert: 1 (4.8%)
Organization	
University: 6 (28.6%)	Industry: 4 (19.0%)
Defense Organization: 1 (4.8%)	Internet Provider: 2 (9.5%)
Government Office: 1 (4.8%)	Security Provider: 7 (33.3%)
Participant Group	
Freelance Workers: 12 (57.1%)	Industry Workers: 3 (14.3%)
University IT/Sec.: 6 (28.6%)	

content. Demographic information on gender, advice generation role, and the organization type is presented in Table 4.1.

4.2.2 Instrument Creation

Our goal in this study was to investigate the processes, decision-making, and challenges that play a role in the creation of general security advice. Based on our research questions, we drafted an initial set of high level questions. Using this draft, we then conducted two practice interviews and one pilot interview; our pilot was with a researcher who has experience writing general security advice. Based on these interviews, we revised our interview guide into three background questions and nine high level questions corresponding to our research questions, each with sub-question-level prompts.

Our interview guide contains questions about processes, decision-making, and challenges, such as *"Can you tell me about how security advice gets made and distributed at your organization?"*, *"Are there particular areas that are prioritized or discussed more in depth within the general security advice?"*, and *"Are there any tasks completed during general security advice creation/revisions that are challenging or time consuming?"* The high-level

version of our interview guide can be found in Appendix B.1; we provide the full version with prompts in our replication package².

4.2.3 Interview Process

Choosing semi-structured qualitative interviews as our research method allowed us to ask broad questions about advice writing and then follow up with more specific questions where appropriate. Once we met participants virtually to be interviewed, we confirmed that they had read and understood the consent form and began recording. We reminded participants of the options to skip questions or terminate the interview, and we gave them a choice of audio or video recording. All interviews were conducted and recorded remotely via Zoom. Recordings were backed up with Open Broadcaster Software (OBS) (OBS 2022). All interviews were conducted in English and lasted between 30 minutes to an hour.

4.2.4 Data Protection

We took multiple steps to protect participants' privacy and data security. First, all participants were pseudonymized. Once interviews were completed, we saved audio recordings of the interviews and had them transcribed by a GDPR-compliant transcription service. Within each transcript, we thoroughly removed all personally identifiable participant data such as names, organizations, and demographic information. We also did not request identifying, confidential, or private information about our participants or their employers in our interviews. We used end-to-end encrypted tools in all of our study communications and data storage components.

4.2.5 Data Analysis

Data analysis was performed through inductive and deductive qualitative coding. We use coding not as a means to an end, but as a strategy to make sense of our data (Elliott 2018). Codebook creation, coding, and discussion of disagreements helped us understand the data, formulate the themes that we describe in our results, and describe advice creation. We created our qualitative codebook based on our research questions, then expanded it with additional codes that emerged through open-coding the transcripts. The codebook was iterated over through weekly discussions with the team and through discussing and

² <https://advice22.netlify.app/>

resolving disagreements between the first and second coder. The high-level version of our codebook can be found in the Appendix B.2. The detailed operationalized codebook is included in our replication package. During the codebook development process, the coders independently double-coded 5 transcripts, with good inter-rater reliability at Krippendorff's $\alpha > 0.75$, and resolved all conflicts through discussion (Freelon a), after which the primary coder coded the remaining transcripts. With Krippendorff's $\alpha > 0.75$ for all transcripts, we are confident that our codebook is stable, represents our data well, and that our coding strategy was sound (McDonald et al. 2019b). Altogether, the coders coded 21 and 5 transcripts, respectively.

4.2.6 Limitations

As with any interview study or self-reporting study, participant responses may be biased (e.g., self-reporting bias, social-desirability bias) or incomplete (Lazar et al. 2017). Specifically, some participants were not able to answer all questions we asked due to either a lack of access to that knowledge or a lack of experience.

Over half of our participants were freelancers (57.1%) who all reported writing advice for company employees in some consultation role. We also do not have detailed data on the audiences beyond what authors reported, though we feel it reasonable to assume only relatively large organizations have employees dedicated to this task. Some freelancers specialize in security advice, while others work on technical writing more broadly. Authors wrote for employees, university students, customers, or users, but in all cases, the authors assumed readers have a “general public level knowledge.”

In theory, it is possible that paid participants would fraudulently participate in interviews. However, for participants recruited from advice websites, we are reasonably certain that they were genuine. For UpWork recruits, we specifically reached out to those who listed relevant expertise on their resumes; since writing is not UpWork's main focus, there is little incentive to fraudulently report this expertise. Participant pre-survey data and interview behavior also matched up. We are therefore reasonably sure that our participants were genuine.

Lastly, any study involving qualitative coding is subject to author biases and different coding strategies among coders. We address these biases in our investigation by first establishing a list of high level coding categories a priori that represented the high level questions that we developed in the creation of the interview, as mentioned in Section 4.2.2. A second

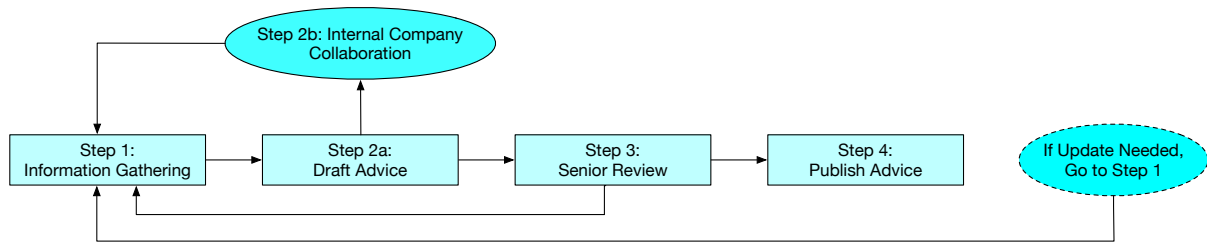


Figure 4.1: How experts write general security advice.

research team member double coded five of the transcripts to ensure that the codebook was able to capture data that reflected our research questions, regardless of the coder.

4.3 Results

In this section, we present our qualitative findings from analyzing the in-depth perspectives of advice writers during general security advice creation. We use participant quotes to represent in their own words how participants answer our interview questions, and ultimately our research questions. We present exploratory findings for understanding the processes, decision making, and challenges encountered by the authors who write general security advice. Such volunteered information includes the company, target audience, and advice generation role they assumed while writing general security advice.

We find that participants followed a common advice creation for advice writing, found in Figure 4.1. This four-step process reflects advice-writers gathering information, drafting advice, sending the advice for review, and then publishing the advice, with options for iteration and further information gathering and collaboration in each stage. We also find that authors primarily prioritize and revise their advice content based either on current security trends or in response to security incidents. We organize the remainder of our results around this process. In Section 4.3.1, we describe how participants gather information for their advice content. Next is Section 4.3.2, where we explain the decision making that advice authors make when drafting and revising the advice. Section 4.3.3 details how advice authors collaborate with different internal company departments on the drafted advice before it is reviewed by senior-level employees and the company’s legal department, in the case the legal department was involved. Lastly, Section 4.3.4 reports challenges participants mentioned for writing general security advice and also what improvements they stated

could help general security advice writing.

4.3.1 Information Gathering

Here we explore findings from the first phase of advice writing, and information gathering. In this phase, authors are simultaneously gathering specific pieces of information (e.g., “prefer long passphrases”) and establishing the scope of the advice more generally (e.g., “we should discuss password strength”).

Theme 1: Advice writers research their environment to scope their advice. Some participants indicated that clients or stakeholders had already defined the scope of their advice documents. In the majority of cases, though, participants were left to figure out what advice was needed based on their own research of the client environment or what would be compliant with the organization. In these situations, authors develop a conceptual model that identifies what issues need to be addressed within the advice. This model is based on the organization, technology, and problems an organization currently faces (e.g. what areas where they are weak in security). Two broad approaches emerged from the interviews: holistic review and gap analysis. The primary difference between them is whether the advice is being written “from scratch” or to supplement existing advice.

As an example of holistic review, F020 explained they begin by determining “what are the devices, connected devices, the architecture and design of the network as well. How the company or how the devices are connected to each other and how the information is being transferred and sent between all the devices.” From there, security advice is drafted for the elements of the system. In a gap analysis process, authors compare their architecture and operational environment to the advice available, and where advice is not present for an area that motivates additional material:

U019 *Honestly, I report directly to the CISO. We meet every week for an hour at least, and we try to stay in lockstep about where the gaps are with what our incident response and governance compliance, and all those other teams are doing that can be addressed through getting educational materials out there in front of people.*

Theme 2: Advice writers base their own writing on multiple distinct external sources. 20 out of 21 participants stated they refer to an external source for sample information to include in their advice. These external sources may be regulations, technical standards, or industry or government agency documentation. Participants reported using multiple types

of source for their content. 12/21 participants refer to legal regulations, specifically privacy regulations like GDPR(7/21 participants) and HIPAA(4/21 participants). One participant said about regulations they refer to:

F008 *I would even say GDPR, where it clearly mentioned ‘portal’, ‘what is personal data’, ‘what kind of controls’, ‘how consent should look.’ Because, so easily, every company after the Internet boom — every website — was collecting data randomly from users without their consent.*

Participants specifically mentioned reviewing ISO standards (10/21), NIST publications (9/21), and PCI-DSS (8/21). In some cases, participants’ employers may suggest that the advice they publish should comply with regulations or standards the company adheres to. In others, the participants rely on these documents primarily to influence content:

F006 *Normally, most of the companies we experienced are starting, and for certain companies, we normally recommend starting with NIST and ISO; they are well known, and their resources are well known, and they are well used.*

Authors also seek out external sources because the target audience may need to understand how to use specific software or how to mitigate a specific security issue.

Sample advice in online web postings from government agencies was also referenced by participants as additional guidance for the advice they write, as one participant states:

F020 *I use also some additional resources, such as NCES.ED.gov. This is the link that I also found on some extra information about security agreements, about some templates, for example.*

In aggregate, our participants cited a total of 20 *distinct external sources* for their own work. Overall, these sources tended to be authoritative, so the content is likely correct. However, if these “upstream” standards documents were not properly scoped or written solely for technical experts, those issues may propagate “downstream” to general advice.

Some sources (e.g., GDPR, NIST publications) saw wide usage by a plurality of participants. However, of the total 20 cited external sources, only 5 were cited by 6 or more of participants.

4.3.2 Advice Drafting and Decision-Making

In the second step, advice writers draft advice based on gathered information and specific decision-making. Decision-making that impacts the advice content includes considering what areas of advice to prioritize, perceived responsibility authors held in advice writing, and addressing the usability of the advice.

Theme 3: Advice writers prioritize content in response to specific incidents or current trends. 13 participants indicated that specific security incidents or current industry trends were key prioritization factors. These participants form half of the participants in each of our four advice generation roles and form the majority of participants in each of the six organization groups. F006 describes how remote access became important during the COVID-19 pandemic:

F006 *From my experience, most of the companies right now have employed remote work for their employees... They have a higher risk of having their data breached or compromised. So for me, what I would say, the remote access policy is the most important in this area, and we have to look into that because it's easy to compromise and very hard to detect what has happened.*

One participant described how incidents at other organizations led to advice creation:

I002 *Sometimes if they give advice, it's based on something that's going on. For example, in a ransomware attack, they say, "Okay, universities are confronted with this type of attack, we should be careful with this."*

Advice creation may also correspond to events within the organization.

U007 *I would say anything time critical is going to be prioritized, so if a change is happening and there's a deadline by which a user is going to need to make a change in accordance with whatever it is that's occurring, or people need to know about this change before it occurs, something like that is certainly going to be a priority.*

Theme 4: Advice writers most commonly cover online fraud and password security. While prioritizing advice on current trends or incidents was common, several specific areas were highlighted by participants. 10 participants indicated a priority on online fraud (e.g., phishing, social engineering, identity theft, email scams). On online fraud, U018 said:

U018 *I've seen a lot of advice that we've been putting out regarding job scams or email scams. Phishing is a big one that we've put out a lot of general guidance on. Those are the only big ones that really come to mind is the job scams and the phishing.*

Five participants mentioned password security and authentication. On authentication, I002 said:

I002 *And then, of course, password security is also a very important topic on everything that has to do with password security, like use of password managers, multi-factor authentication.*

Theme 5: Advice updates are reactive. 16 participants reported updating advice after new security trends or incidents become prominent. This theme of revising advice to reflect current trends or security incidents was also mentioned by at least half of participants within all six organization groups. A theme that indicates prioritizing content over novel or recurring advice topics. A majority of participants within the analyst, awareness expert, and security expert advice generation roles responded similarly. On reacting to new security incidents, U018 said:

U018 *Any new information that we find we like to provide to the community so that they're aware of the new ways that these scammers or the phishers or the bad actors are trying to get to them.*

Theme 6: Advice writers cover a wide variety of less-common topics. Outside of online fraud and password security, participants mentioned prioritizing 12 other areas of advice, but no more than 4 participants mentioned any particular topic. Participants mentioned areas like remote access policy management, frameworks, organizational policies, incident response, and risk assessment. These trends concur with prior work that indicated a lack of consensus on advice prioritization (Redmiles et al. 2020).

A potential explanation may be that organizations have different advice needs, and our interviews support that interpretation. While fraud advice was the highest mentioned prioritized advice, it was only mentioned by half of the participants within only three different company types, respectively. Every participant representing either a defense company or internet provider mentioned prioritizing fraud advice, and five out of six total university workers mentioned prioritizing fraud advice. The rest of the advice areas are sparsely mentioned among the different participant sub-groups. This finding may also be

caused by our Theme 5 findings in that new advice topics are constantly being addressed, given whatever security incident or trend is current.

Theme 7: Advice writers rarely curate advice by intentionally deprioritizing topics.

Most participants did not provide significant responses when asked what areas of security advice were *not* prioritized or why. Four participants said advice for obsolete or deprecated technology would be removed or deprioritized. When asked, F016 replied: “*There’s a lot of old depreciated functionality in Microsoft Windows.*” (F016). Others mentioned deprioritizing impractical or overly technical advice, though in at least one case this deprioritization was reluctant. For example, I003 mentioned encryption advice as a topic to deprioritize given being “too technical” for general security advice.

Theme 8: Advice writers consider usability, but without a consistent or systematic methodology. 20/21 participants mentioned an attempt to make their general security advice usable. However, participants mentioned 9 different methods to address usability, none of which were mentioned by more than 8 participants. 8 participants stated they simplify the technical language of the advice so that the general public can understand the advice. Participants also mentioned using visualizations and graphics to enhance usability:

I002 *Our job is to translate this to something less technical and comprehensible for a broad public. Make it sometimes also a bit more visual — more attractive for users.*

Six participants determined if advice was usable by considering how they themselves would follow the advice.

I004 *One of the things that we look at as we’re writing the advice, how would we implement it? If we can’t figure out how to implement our own recommendations within our own teams...how could we possibly expect people to be able to implement this?*

Some participants considered usability but without a specific method for writing or evaluating advice usability.

U018 *It’s not really a process per se that is implemented. It’s just something that we keep in mind to make it as user-friendly as possible.*

Theme 9: Help desks are considered a backstop for unclear advice. 20 participants stated their organization had a team (such as a help desk) that could address users’ security questions. U007 noted that they assumed that if advice is unclear, the help desk would correct the issue.

U007 *Our expectation in every case is if the user is looking at instructional pages and they're confused about what they mean, or basically confused about anything that they see on the central IT website, that they would contact the help desk.*

We believe that this perspective may be optimistic about users' likelihood of asking questions before engaging in an unsafe action, and in any case this would be an interesting perspective for future work. On the other hand, if help desks receive the same questions frequently, it may be an indicator to authors that they should update advice on a topic.

Theme 10: Advice writers claim a wide range of responsibilities when writing general security advice. Prior work established a mismatch between users' and manufacturers' expectations of each other in smart homes (Haney et al. 2021). Inspired by this effect, we asked participants about how much responsibility they or their company bear in advising users of secure practices. Overall, participants gave answers ranging from high levels of responsibility to virtually no responsibility. We also noticed that responses differed between the participants' roles, though we do not claim a literal correlation because this is an initial qualitative study. One of the university participants noted they take extreme ownership of their users' security education, and they stated their team's goal is to: "*help them, guide them, and educate them, and basically be a partner with them*" (U018). On the other hand, a freelancer gave a different perspective on perceived responsibility:

F011 *Well, we give advice for the sake of advice. We want to be sure that we are not promoting what we do or represent. We are not trying to sell, like force you to patronize or do business with us*

An industry worker mentioned assuming varying levels of responsibility depending on the content:

I004 *It depends on what we're writing and why. If I'm writing the security advice or scoping the payment card industry data security standards audits, I'm just going to lean towards every time writing advice that would limit the scope of our environment because obviously, that makes it easier for us to pass the audit. If I am writing advice for, say, being ready to do something with data privacy, for instance, marking data as highly confidential things like that. I'm going to be as broad as I can be because I want everything protected*

Overall, we recorded seven different categories of responsibility suggested by participants, though none were mentioned by more than six participants. Other levels of perceived

responsibility included writing advice to comply with standards, motivating changes in security behavior, or going beyond documents to offer security workshops.

4.3.3 Collaboration and Review

In step two of the writing process, advice writers collaborate with internal company departments who review the advice. Then, in step three, the advice is sent to senior-level employees for final review and approval. In both steps, advice writers revise until the requested revisions are approved by the relevant stakeholder. Otherwise, the advice is approved and then published in step four.

Theme 11: Advice writing is distributed and collaborative. 16 participants stated there were multiple writers who worked on content, and they stressed the importance of multiple writers to lessen workload and include multiple perspectives:

U010 *Currently, I tend to get most of the cybersecurity writing assignments in our group. I wouldn't necessarily say all, because we do have a focus on trying to develop bench strength, and within our group there are also people who are responsible for the communication regarding specific IT services offered by departments.*

18 participants mentioned collaborating with internal company departments. Security (12/21), marketing and communications (7/21), and human resources (HR) (3/21) were the most mentioned departments for advice collaboration. Participants noted that the mix of security experts and non-experts helped create content that is technically accurate and understandable to a broad audience:

U007 *The central IT communications group (none of them have been members of the information security group) [works] closely with information security. So if they're writing about something that's a security issue, they're going to be corresponding with one or more people within information security making sure that they have the details right in their write-up. Or they may start with a couple of paragraphs that someone in information security supplied, and then they'll write their page around that to make sure that they're getting the technical details right.*

Theme 12: Advice is routinely reviewed by senior personnel. 17 participants mentioned submitting proposed general security advice to senior level employees (management or advisory board) for review, feedback, and approval. The university employees and industry

workers we interviewed keep their advice within their own group before it is sent to their own management who then approves the advice. Freelancers writing for other organizations submit their advice to the management of their client company for review and approval. For example, participant F014 mentioned review by C-level executives.

Theme 13: In-house legal counsel can be heavily involved in advice creation. We reasoned at the beginning of the study that one cause of the overall lack of prioritization of security advice might be organizations writing comprehensively to limit liability rather than focus on the most likely issues. Therefore, we specifically ask about the involvement of counsel.

7 participants confirmed that their legal department was involved to ensure that the advice met certain legal standards and was up-to-date with the current law. As F011 explains:

F001 *We believed that being a good lawyer does not mean being a good security expert, ... so [lawyers] just review. If they feel something should be removed on legal grounds, they advise us... and if they feel that we need to include some things based on legal grounds, they let us know. We include those things and then send that back to them for review, and then we go back and forth until they are satisfied with the documents.*

Another seven participants stated their legal was occasionally involved depending on the content or intended audience:

I003 *If it comes to data protection specifically or a GDPR specifically, yes [legal is involved], because they would be the experts. For the rest, no.*

The remaining seven participants indicated that they were either uncertain of the role of legal counsel or were certain that legal counsel was not involved.

While legal counsel indeed influences content, we conclude that the extent of their influence does not explain the breadth and variety of security advice reported in the literature.

4.3.4 Reported Challenges and Improvements

Theme 14: Advice writers struggle to scope advice for broad audiences who lack fundamental security knowledge. 12 participants specifically mentioned difficulty in identifying not just the necessary topics for their intended audience, but also explaining relevant solutions for diverse technical settings. Simplifying advice content was recorded in responses

by participants who mentioned it as a possible method to improve future general security advice. Throughout the study, participants gave examples of how both the intended audiences and teammates they worked with during advice writing came from different backgrounds and therefore all have different levels of security awareness. Therefore, it is important that general security advice be simplified for all intended audiences:

U019 *I think not underestimating your audience, trying to empathize, trying to put things in terms where they understand that what we're helping them with is really the thing that they want the most.*

This challenge is especially seen in advice for employees who need to use software but lack necessary security awareness. Specifically, writing advice content that accounts for accurate assumptions about the intended user's security knowledge. One participant stated it is easier to advise more experienced clients who have security knowledge than those who are less experienced. Another participant mentioned that

I002 *The problem internally was I think mostly that our IT teams supposed that everyone knew that we had a password manager, and they knew how to use it. But basically, this wasn't the case because a lot of people — I think also a lot of new people working for the organization — didn't know that it existed*

Theme 15: Advice writers value direct security training, despite its costs. Participants perceived that rectifying gaps in fundamental security knowledge is difficult:

F006 *The most time-consuming task is when a company needs to train their employees, which takes a little time to train and give awareness to the employees.*

They also still recommend direct training.

F011 *We believe in constant improvement. We believe in trainings. We believe in our teleconferences. We believe in individual investments in their whole training. So we encourage our team members to learn more. As a company, we try to find where we can get the kinds of trainings, conferences, events and all of that so that we can get updated on the current trends and security.*

One participant noted an alternative approach to traditional trainings:

U009 *During the Cybersecurity Awareness Month we have games (particularly online, given the COVID). I'm astounded at how many people reach out and want to play these games for \$25-\$50 gift cards. So provide more games to attract people and use that to ask them questions. Perhaps one game session focuses on multi-factor, and another game focuses on strong passwords.*

Theme 16: Participants recognize the need for proactive updates. Participants also desired to revise or audit published advice on a regular basis, as opposed to strictly reactively as discussed earlier.

U007 *Somebody needs to be looking through the pages and making sure that they're still relevant and that they still have current information, and I think that's an area that we're not really that good at.*

Theme 17: Collaboration and review leads to delay in publishing advice. We previously discussed how authors credited collaboration and review with leading to more readable and appropriate advice. Participants mentioned that collaborators who either are not consistent in their practices or do not meet important guidelines when writing the advice are challenging to work with. 6 participants mentioned that collaboration leads to delay because it requires the time of multiple busy parties. Time from senior stakeholders is even more difficult:

I004 *A lot of times, those stakeholders are upper management. It's hard to get on their schedules. So there's always room for improvement in that. Sometimes the process takes too long because you're literally waiting for a day when you can get four people in a room together, and it's two weeks out. So there would be room for improvement there. Maybe that's top-down buy-in where they say this is more important than anything else; make time for it, which only happens after a breach. I would say those are two areas that would make things easier.*

4.4 Discussion

In this section, we contextualize our results with prior literature on advice prioritization, procedural decision making, and perceived responsibility in end user security. We then discuss how these findings can promote better practices for curating general security advice with methodological recommendations for both advice writers and their organizations.

4.4.1 Identifying Lack of Consensus

Advice Prioritization: Participants prioritized their advice to reflect current security trends or respond to security incidents during advice construction and revision. Prioritized advice topics experience variable attention over time and may undergo fluctuating cycles of prioritization. Similarly, advice revisions exhibit a reactive manner in an attempt to constantly keep up with the latest security incidents and trends. While it may seem appropriate to prioritize content like this, it leads to a possible overproduction of advice on numerous security topics. We see this in Theme 6 as participants mention a total of 14 topics they prioritize in their advice writing.

Differences in prioritized advice topics among our study population may also be likely contributed by the differences in specific target audiences, roles, and organizations. However, even participants among the same groups sparsely agreed on which specific topics of advice they should prioritize. Rather, they instead looked to whatever novel security threat they determined they needed to cover. Relying on the latest threats and trends for advice writing also makes it more difficult to determine which security advice should *not* be prioritized. Outside of not covering obsolete or impractical advice, advice writers rarely provided significant responses to how they would deprioritize security advice.

We believe the reluctance to curate or deprioritize content partially explains the advice prioritization crisis documented in prior work. To borrow a common expression, “if everything is a priority, nothing is.” Practitioners recognize that the attack surface for modern computing is vast and ever-changing. However, they may fail to account for the effort and opportunity cost to users caused by comprehensive advice. Our findings on the curation of security advice from advice writers add context to a continued theme from prior work indicating a lack of consensus on which general security advice should be prioritized (Redmiles et al. 2020). Our results show that content covered in security advice is not curated to cover perennial topics, rather it is curated to cover many novel topics. Focusing on novel threats instead of perennial threats for security advice may contribute to overwhelming end-users with security advice they do not need. This implies that much general security advice found online may either be outdated or less relevant than when it was first written. Users of varying levels of security experience are left on their own to distinguish which security advice they actually need or is still important. While security experts are better equipped to make these important choices, end users lacking proper security awareness are less likely to understand the distinction between outdated and relevant security advice.

This increases the number of security topics that end-users have to read through in advice and determine which advice they should prioritize. We make recommendations for a more proactive approach to advice and improving the lack of consensus for prioritizing general security advice in Section 4.4.2.

Procedural Decision Making: A lack of consensus in procedural decision making was also prevalent in our findings. This is first observed among our Theme 1 and 2 findings which describe how information is gathered for their advice. We discovered that advice writers experience challenges in identifying key aspects of the scope of the advice they are tasked to write for their intended audience. This is a critical challenge given that most participants in our study state that information gathering is their first task in writing advice and sets the foundation for the scope of the content. If advice writers then make decisions on prioritization given an insufficient foundation, their advice will experience variable prioritization, given their inability to consistently define a scope for their advice. P009 stated they would prefer their advice to “target messages to students. It’s a challenge. What we create is available to them, but how it’s delivered and where it’s delivered should be better targeted.” Participants among all of the recruited groups experienced this challenge regardless of their organization or intended audience. General security advice affects end users in universities, organizational employees, and many other types of audiences who lack adequate general security awareness. A lack of consistency in properly identifying how general security advice should be written leads to a lack of consensus in advice prioritization from experts, which then trickles down to the lack of advice prioritization among general users.

Advice writers also refer to multiple distinct sources of sample advice that include any legal regulations, technical standards, or other organizational entities. In total, we identified 20 different external sources that participants mentioned they use to influence their general security advice. Of the 20 different external sources mentioned, only four were mentioned by at least 30% of participants. Similar to how this work and prior work (Redmiles et al. 2020) demonstrate a lack of advice content prioritization, there is also a lack of consensus among organizations and advice writers on which external sources to pull sample advice from. We see this lack of consensus among participants of the same recruitment group and also between participants of different recruitment groups. Specifically, we observe differences in external sources cited for influencing advice content even among participants who share both the same intended audience and the type of organization they worked for. An upstream

usage of distinct external sources and an inconsistent foundation for gathering information for advice add more clarity as to why security experts differ in security advice prioritization. As suggested in prior related work (Haney et al. 2021), we discuss the importance of both advice authors and organizations to formulate standards to consistently curate perennial topics for security advice in Section 4.4.2.

Lastly, it is evident there is no consensus on agreed-upon methods in which advice authors consider the usability of their general security advice. Also, it is unclear if participants are performing appropriate usability checks in their advice for their intended audience. This lack of consensus for methods in considering the usability of general security advice construction may be due to the lack of experience or knowledge about usability from both the authors and organizations. Another possible factor could be the pressure to release advice for a client within a specific time frame, a challenge that is seen more often for industry or freelancer workers. This increased pressure to meet deadlines to produce content may come at the expense of thoroughly considering the usability of general security advice. Regardless, a lack of consensus on which usability methods to implement to write general security advice can create advice of variable degrees of usability. Prior work has shown that end users follow or reject advice based on perceived cost and benefit analyses of the advice (Herley 2009; Fagan and Khan 2016; Christin et al. 2011; Beautement et al. 2008). This adds another burden that end users have to consider when deciding what general security advice they should prioritize. On top of deciding if advice presented to them is relevant to their needs, they also have to consider if that advice is worth implementing given the opportunity costs of implementing the advice. We recommend that specific usability tactics should become standards agreed upon by experts and authors of general security advice in order to lessen this burden. We explain this in further detail in Section 4.4.2

Perceptions on Security Responsibilities: Previous work by Haney et al. (Haney et al. 2021) identified an interdependent relationship within user perceptions of the responsibility of smart home device privacy and security between three actors, namely the smart home device end users themselves, device manufacturers, and third parties such as government or regulatory bodies. In that paper, it was reported that users based their actions on that perceived interdependent relationship and therefore would not consider themselves the sole protector of the security of their smart home devices. However, manufacturers and regulatory third parties do not always act on this interdependent relationship, thus there exist gaps in understanding of the responsibilities for each party. In our work, we discov-

ered several different responses for how authors of general security advice assess their perceived responsibility in advice writing. Some participants only wrote advice to comply with standards or requirements their organization enforces whereas other participants emphasized a need to educate their intended audience to develop better security decision making habits. Overall, there is no agreed upon consensus for what levels of responsibility that authors of general security advice or their organizations should assume in assisting their target audience. Our findings support results from previous work and highlight gaps in responsibility assumed not just between the general public and entities providing content, but also between the parties responsible for generating general security advice for users. Gaps in perceptions of shared responsibilities among the experts and end users fall further than just for smart home users, but also for both the general public and organizations who seek assistance with general security. A lack of agreed upon consensus on perceived responsibilities is apparent within both end users and the experts. This leads to increased confusion about which parties should be responsible for mitigating what issues or making security decisions for end-user software and technology. In Section 4.4.2, we add onto recommendations from previous related work (Haney et al. 2021) and make suggestions for explicit responsibility establishment for future general security advice creation.

4.4.2 Methodological Improvements for Advice Writing

We identify areas of improvement for general security advice construction by analyzing the current state of advice construction from the lens of the writers. We suggest methodological improvements for general security advice construction based on our current findings and findings from previous related work (Haney et al. 2021; Redmiles et al. 2020; Cornelissen 2020).

Develop the Domain of General Security Advising

A lack of consensus across multiple areas in the general security advice writing process indicates that its domain is not fully developed. We describe six domains of focus for professionals to consider when writing general security advice.

Resources/Technology: The biggest challenge participants mentioned when writing general security advice was defining the scope of advice content and how the advice can be

broadly applied to all intended targeted audiences. P004 and P021 both advocated for the creation of an open source based repository to act as a research forum for advice writers. Both participants recommended that such a system should contain organized information about current general security questions or issues and that this system can be queried or allow open discussion between advice writers. Such a tool could be used by advice writers to both better discover what security topics need advice on and collaborate with other advice writers to come up with implementable solutions to communicate with their target audiences. While this is one idea of tool creation, future research may investigate the creation of new tools to help authors of general security advice identify advice content. Advice writers then would not have to rely on waiting for an incident to happen or a new trend to become popular in order to generate ideas for general security advice.

Relationships: Multiple parties are involved in discussing and reviewing general security advice before it is published. Understanding differences in viewpoints and requirements among organizational parties has been studied at a broader scope for internal corporate communications (Cornelissen 2020). Critical challenges described in such work emphasized the importance of addressing communication related management problems between management and communications practitioners. Similarly, we recommend that advice writing parties each receive clear definitions on their responsibilities and roles during advice writing to reduce confusion among everyone involved.

Participants also mentioned that meeting with every party involved in the writing process can be time consuming since different parties (e.g., marketing, communications, security) have different schedules and duties to adhere to. Authors of general security advice should consider adopting a formal schedule and process that is agreed upon by all collaborating parties. This may help decrease the amount of time waiting for collaborating parties to meet and discuss advice content and how it should be implemented.

General Security Focus: Improving the security culture of both the intended audience and parties who write general security advice was recommended by participants as a means of improvement. Security awareness games, workshops, and other events to keep authors of general security updated on security trends help them stay connected to what issues are affecting their target audience. Providing the same programs for general users is also helpful for them to learn general security advice in a non-conventional way and should be considered by experts who write general security advice.

Content Improvement Metrics: Agreeing to a set of usability practices to make general security advice more usable can consist of the following: advice visualizations (e.g., diagrams, images, media, etc), and simplifications of overly technical words or phrases or templates to organize the advice content.

Community Support for Advice Writing: Establishing a community for advice writers to collaborate on ways to address common security threats through advice would greatly help writers in earlier stages of advice writing. Such a community can communicate by sharing best practices, sample advice from experts, or even recommendations for non advice-based approaches (e.g. games, workshops). Methods to evaluate the effectiveness of published advice over time can also be agreed upon by a community of advice writers. Academic researchers should also be involved in community collaboration in advice writing.

Human-Centered Engagement in Earlier Writing Stages: Advice writers in our study rarely mentioned user interaction with their intended audience as a means to generate content. Earlier stages in the writing process would benefit greatly from engaging directly with their intended audience to learn about security problems they may encounter. These direct interactions can help inform writers on which topics to prioritize, as well as how understandable or actionable their advice is. Writers should also gauge their users on whether the volume of advice is too high.

Proactive Advice Updates and Curation

Proactive Advice Updates: Implementing a proactive manner of updating or reviewing general security advice better ensures the advice is up to date. Participants mentioned performing more frequent check ups or audits of general security advice helps maintain the relevancy of the advice. Without consistent content audits, there increases the chances of advice becoming stagnant or not reflective of the current environment. Therefore, adopting a proactive approach to reviewing general security advice on a timely basis prevents the presence of outdated advice.

Establish a Set of Agreed Upon Standards for Advice Curation: We advocate there be a consistent standard to determine perennial areas of general security advice to cover. We say

a set of standards since no one standard can be broadly applied to all advice of all intended audiences. Therefore, we suggest advice authors and industries communicate both what advice should be perennial and what advice should not be prioritized. Also, we suggest the research community investigate further what security advice end-users actually claim they need and if they are receiving that advice now.

4.5 Conclusion

In a semi-structured interview study with 21 authors of general security advice, we analyze the processes, decision making, and challenges that experts face when writing general security advice. We corroborate the lack of consensus on security advice as well as responsibility assignment from prior work. Our contribution gives insights into the context and reasons for this lack of consensus: advice writers struggle to define the advice scope and prioritize the information necessary to write advice, and must prioritize time-sensitive events over curating perennial advice. Based on our findings, we provide recommendations for how general security advice authors can better develop the domain of general security advice writing, implement proactive approaches towards writing and revising general security advice, and establish a set of agreed-upon standards for advice writers to reference when curating general security to end users. Addressing the lack of agreement on how general security should be advised from both the end user and advice writer side may improve the relationship between both parties in general security advice prioritization and perceived responsibilities.

CHAPTER

5

IT SHOULD BE EASY BUT... NEW USERS' EXPERIENCES AND CHALLENGES WITH SECRET MANAGEMENT TOOLS

5.1 Introduction

Software developers need to store software secrets (e.g., passwords, API keys), which grant access to sensitive or private information (e.g., account access, company bank information). Without a proper secret management approach in place, developers may hard-code secrets into source code, public repositories, or configuration files. A highly recommended option by industry, latest guidelines, and academic research (Basak et al. 2022, 2023b; Krause et al. 2023) are secret management tools (SMTs). These can securely store sensitive developer secrets in a centralized location. However, even with the existence of SMTs, developers

still leak secrets (Katz 2023; Basak et al. 2023b; Krause et al. 2023; Dahlmanns et al. 2023; itnews 2023), which can lead to a wide exposure of sensitive information (Mcdaniel 2023b; Abrams 2023; Meli et al. 2019; Burgees 2024; Jackson 2023).

Like most software, SMTs come with online public documentation that instructs developers on how to store, manage, and distribute secrets. Prior research efforts have highlighted issues within software tool documentation that affect developers' abilities to use the tools to perform software development related tasks (Parnin et al. 2012; Nassif and Robillard 2023a; Robillard 2009; Ernst and Robillard 2023; Acar et al. 2016, 2017a; Fourné et al. 2024). Limitations for software development tool documentation may extend to SMT tool documentation, yet the extent of the impact of the quality of SMT tool documentation on developers' experiences in using SMTs is unclear. Prior research shows that SMT adoption is widely a burden for developers, largely because of setup and learning constraints during initial adoption of the tools (Basak et al. 2023b; Krause et al. 2023). Ultimately, we do not know how effective current SMT tool documentation is towards helping new users learn how to use SMTs to centrally secure their secrets.

To understand the experiences and challenges new users face when using SMTs, with a specific focus on tool documentation, we ran a mixed-methods laboratory study with 21 Computer Science (CS) Master's students with prior industry or academic experience with managing secrets. We investigate the following research questions:

RQ1. Can new users effectively store, access, and inject a secret using a SMT within one hour?

RQ2. What factors encourage or inhibit SMT usage success?

RQ3. Does the choice of SMT impact tool usage success?

We observed participants using one of three cross-cloud SMTs: HashiCorp Vault Secrets (HashiCorp 2023), Infisical (Infisical 2023), and Doppler (Doppler 2023). With the assistance of SMT tool documentation, participants performed two tasks that involve managing a hard-coded secret: Access a stored secret from the command line, then inject the secret into a local Python application. Following each task, we interviewed participants about their experiences and challenges using the SMT and tool documentation.

We find that the quality of available tool documentation for SMTs drastically impacts participants' ability to effectively use the tools, which ultimately affects the ability to se-

curely manage secrets. Major limitations in SMT tool documentation that negatively impacted SMT usage were the following: Insufficient information for relevant command line arguments, lack of relevant set of use cases in examples provided for command line interface (CLI) commands, and inconsistent versions of descriptions for CLI commands maintained across documentation. Participants also reported a lack of clarity when using the tool documentation interface to find technical guidance such as specific commands or debug information. When encountering limiting factors in SMT tool documentation, participants often deviated from tool documentation to access secondary sources and attempt workaround methods. However, participants largely reported that secondary sources were less helpful for their specific needs and therefore experienced more time and effort for troubleshooting and exploring solutions. While we noticed observable differences in task completion times between SMTs, all were functionally indistinguishable towards task success. Our work can serve as a reference for SMT vendors who need to create effective tool documentation and more user-friendly tool interfaces for their intended audience, regardless of their technical expertise, to enable broad, secure adoption of effective secret management tools.

1

5.2 Methodology

We conducted a mixed-methods laboratory study on SMT use with 21 Computer Science Master's students in July 2024. Participants were observed while using SMTs, with the assistance of publicly available tool documentation from the SMTs, to perform two tasks: Task 1 involves storing and accessing a secret, Task 2 involves injecting a secret into a local application. After each task, we interviewed participants to learn their experiences and any challenges they faced while completing each task. In this section, we explain the methodology of our study.

¹ Text of this chapter is reprinted with permission from Lorenzo Neil, Deepthi Mungara, Laurie Williams, Yasemin Acar, and Bradley Reaves. It Should Be Easy but... New Users' Experiences and Challenges with Secret Management Tools. Manuscript under submission.

5.2.1 Recruitment and Eligibility Criteria

Our population of interest are developers with academic or professional experience with managing secrets for coding projects. For feasibility, we focus our recruitment towards a sub-population: CS Master's students at our university. Many local CS Master's students either have experience managing secrets from their coding projects, prior employment positions, research projects, or academic courses, and are therefore an appropriate sample for our study.

We recruited CS Master's students through a public email forum offered by our CS graduate office. We described our research overview and goals, and provided a link to our Qualtrics (Qualtrics 2022) screening survey for participants to indicate interest. We required participants to participate in person at our research campus and have either prior experience or general understanding of the following: Linux or MacOS terminal commands, Python file execution, environment variable usage, and secret management, all essential for performing our research tasks. Participants were only eligible if they met all of our criteria. Once eligible participants filled out our screening survey, we sent them a link to our Qualtrics (Qualtrics 2022) informed consent form. Our informed consent form asked for consent to participate in person and agree to be recorded through both video and audio. Participants were compensated \$30 per half hour for their participation in the study.

We ended participant recruitment when we reached saturation (Saunders et al. 2018). When we asked participants about their prior experience, we did not collect sensitive details (e.g., company names or position titles). Participants only shared information they felt comfortable sharing. Table 5.1 shows the breakdown of our participants' prior professional experience, academic experience, gender, and the SMT they were assigned to use in the study. All but one of our participants had prior experience with managing secrets in a company, either through a prior internship or full time employment. Nine participants held a combination of both professional experience and academic coursework experience with managing secrets. They named passwords, tokens, API keys, and user credentials as examples of secrets they were tasked with managing in prior experiences. Participants mentioned using the secret management tools Amazon Web Services (AWS) Key Management Service (AWS 2023), Azure Key Vault (Microsoft 2023), and Google Cloud Platform (GCP) (Google 2023).

Table 5.1: Participant demographics and SMT Assignment.

Participant No.	Professional Exp.	Academic Exp.	Gender	SMT Assignment
P01	Internship	Coursework	Man	HCP Vault
P02	N/A	Coursework	Woman	Infisical
P03	Internship	Coursework	Man	HCP Vault
P04	Full-Time	N/A	Man	Infisical
P05	Internship	Coursework	Man	Infisical
P06	Full-Time	Coursework	Man	Doppler
P07	Full-Time	Coursework	Woman	Infisical
P08	Full-Time	N/A	Man	Doppler
P09	Full-Time	N/A	Man	HCP Vault
P10	Internship	Coursework	Man	Infisical
P11	Internship	Coursework	Man	HCP Vault
P12	Full Time	N/A	Man	HCP Vault
P13	Full Time	Coursework	Man	Doppler
P14	Internship	Coursework	Woman	Infisical
P15	Full-Time	N/A	Man	Doppler
P16	Full-Time	N/A	Man	HCP Vault
P17	Full-Time	N/A	Man	Doppler
P18	Internship	N/A	Woman	Doppler
P19	Full-Time	N/A	Man	Doppler
P20	Full-Time	N/A	Woman	Infisical
P21	Full-Time	N/A	Man	HCP Vault

5.2.2 Study Design

SMT Choice

We chose the following three cross-cloud SMTs: Infisical (Infisical 2023), HashiCorp Vault (HCP) Vault-Secrets (HashiCorp 2023), and Doppler (Doppler 2023). We chose cross-cloud SMTs since they do not require access to a specific cloud ecosystem. All three SMTs offer public online tool documentation and free versions of their tools. These three specific cross-cloud SMTs were also three of the more common cross-cloud SMTs (Matsiako 2023; g2 2023). All three SMTs provide a web user-interface (UI) and installable command-line interface (CLI). The web UI allows users to create free accounts and store secrets. Secret storage dashboards allow users to create and manage projects that store their secrets. From the CLI perspective, users can access secrets from their web secret storage projects and distribute them across local applications without the need to store secrets locally.

Study Protocol

Our study consists of a combination of both a between-subjects and within-subjects research design Budiu (2023). Each participant met with a researcher on campus and performed two research tasks with only one of the three SMTs. Therefore, seven participants performed two secret management-related tasks with each SMT, totaling 21 overall. We present a table that lists which SMTs participants used for our study in Table 5.1 and the study protocol details can be found in our replication package².

We borrow inspiration from related work by providing participants with a scenario-based description for each task (Acar et al. 2016; Oh et al. 2024; Bai et al. 2016). Each scenario imagines the participant working for a software company with the goal to use the provided SMT, with respective tool documentation, to manage a hard-coded secret within a company Python file shown in Figure 5.1. The Python file, `SecureSecret.py`, consists of one or two lines of code for each task. One of the lines of code for Task 1 includes the hard-coded secret `CompanySecretToken`. Participants were provided with a MacBook Pro which held the Python file in a Visual Studio Code (VS) Text Editor work space (Microsoft 2024). Participants also were required to use the terminal prompt within VS to enter SMT-based CLI prompts to complete steps for each task. Lastly, participants were given account access to the web

² <https://smt-study-documents.netlify.app/>

UI for their SMT which held the scenario-based project, “*Company Secret Storage*”, they store `CompanySecretToken` in.

While we described the first task to participants, we provided them links to online tool documentation from the SMT they worked with. The links explicitly provide information for each task within their core CLI commands documentation, quick starter guides, and documentation that covers managing secrets or projects. We present these links to the participant in a Google Doc where we also provide labels for the titles of each link. The list of links we presented for Doppler, HCP Vault Secrets, and Infisical can be found in our replication package. We did this to help the participants reduce trivial lookup time and decrease the disadvantages of finding advice on different SMTs.

Participants were given 30 minutes to complete each task. We allowed stoppages of time when participants asked questions or wanted to take a break. We also allowed participants to skip a task or not answer a question. If participants reached a point in which they felt the SMT tool documentation was unclear or not helpful, we allowed participants to search online for secondary sources for more information. We allowed participants to use any website (e.g., YouTube or Stack Overflow) or medium (e.g., text or video). However, we did not allow participants to use generative AI services (e.g., ChatGPT). We set this restriction since not all participants may be familiar with such services and we wanted to ensure participants have the same resource availability and all know how to use their resources. Relaxing this restriction will be an interesting direction for future work to consider as we discuss in Section 5.4.4.

After participants completed a task, we asked them several questions about their experiences and challenges using the SMT and tool documentation. Participants were also asked a couple of warm up questions before the tasks, which collected their prior experience with secret management. After participants answered questions about both tasks, they were then asked a set of wrap-up questions. We explain in detail our post task interview development and structure in Section 5.2.3 and 5.2.3.

Task Selection

We designed each task to be general and be equally applicable to all SMTs we study. They all can be completed well within 30 minutes. Participants also do not need extensive knowledge of the SMT in order to perform these tasks. With the tool documentation provided to participants, as well as secondary sources, participants were instructed to complete the

```
SecureSecret.py > ...
1  import os
2
3  ##### Python File for Participant Secret Management Tool Research Tasks #####
4
5
6  ### Task 1 Start ###
7
8  CompanySecretToken = "I am a secret token! Please remove me from this code!"
9
10 ### Task 1 End ###
11
12
13
14 ### Task 2 Start ###
15
16 CompanySecretToken = os.environ.get("COMPANYSECRETTOKEN")
17
18 print("Hello, ", CompanySecretToken)
19
20 ### Task 2 End ###
```

Figure 5.1: Screenshot image of the Python file, `SecureSecret.py`, used by participants for Task 1 and 2.

following two tasks:

Task 1: Remove `CompanySecretToken` from `SecureSecret.py`, store `CompanySecretToken` within *Company Secret Storage*, and then access `CompanySecretToken` from the command line using the SMT CLI.

Task 2: Inject `CompanySecretToken` into `SecureSecret.py`.

Task 1 starts with participants removing `CompanySecretToken` from `SecureSecret.py`, as shown in Figure 5.1. For this, we simply have them delete it from the file. Participants then store `CompanySecretToken` using web UI of a SMT in the project named *Company Secret Storage* as shown in Figure 5.2. The last step for Task 1 is to then use the SMT CLI to print the plaintext value of `CompanySecretToken`. A presentation for the number of steps required for this task from the CLI perspective is shown in Figure 5.3. Task 2 involves a process called “secret injection” HashiCorp (2024). For secret injection, a secret is retrieved from the SMT’s central storage, and then passed to an application at run time as an environment variable. Secret injection ensures that the secret is not stored locally or within the application. As shown in Figure 5.1, there are two lines under Task 2 within `SecureSecret.py`. The first

Secrets

Create new secret ▾

🔗 How to use your secrets

Have suggestions for Vault Secrets? [Share feedback](#) ↗





Name	Value	Secret Type	
 COMPANYSECRETTOKEN	Static	  

Figure 5.2: Screenshot image showing the HCP Vault Secrets dashboard for the *Company Secret Storage* project with `CompanySecretToken` stored.

```
1 Command:
2   $ hcp auth login
3 Output:
4   ....
5   Successfully logged in!
6 Command:
7   $ hcp profile init --vault-secrets
8 Output:
9   ....
10  ✓ App with name "CompanySecretStorage" selected
11 Command:
12  $ hcp vault-secrets secrets open "COMPANYSECRETTOKEN"
13 Output:
14  Secret Name:    COMPANYSECRETTOKEN
15  ....
16  Value:          I am a secret token! Please remove me from this code!
```

Figure 5.3: Sample presentation of the list of CLI commands required to complete Task 1 for HCP Vault Secrets.

```
1 Command:
2 $ hcp vault-secrets run -- python3 SecureSecret.py
3 Output:
4 Hello, I am a secret token! Please remove me from this code!
```

Figure 5.4: Sample presentation of the injection run CLI command required to complete Task 2 for HCP Vault Secrets.

line retrieves an environment variable named `CompanySecretToken`. Once the participant completes Task 1, they would have successfully logged in and initialized the SMT with the CLI. Participants can then reference secrets stored in the SMT from the CLI through the usage of environment variables. We take care of this step with the first line of code which creates the environment variable. The second line simply prints “Hello, ” in front of the value of the environment variable we created for `CompanySecretToken`.

After participants complete Task 1, injecting `CompanySecretToken` only requires one command as shown in Figure 5.4. This is common for all of the SMT’s for each Task 2 as each SMT’s injection command consists of a similar syntax which all commonly use the term “run”. The “run” term is followed by two dashes and then the normal start command one would input to run the application in which they are injecting a secret into (e.g., `python3 SecureSecret.py` in our case). The expected output for Task 2 is: `Hello, I am a secret token! Please remove me from this code!`.

Study considerations: We initially chose a Python local file for Task 2 since before our pilot study, none of the SMTs held a specific Python example for their secret injection run advice. We also chose Python since it is the most popular coding language as well (Cass 2024; Jansen 2024; Stackoverflow 2024), and would provide a common language that developers code in already. We carefully designed these tasks to minimize any advantages between the SMTs and their tool documentation. However, after our pilot study, HCP Vault Secrets updated their documentation in their web dashboard to add an example of how to complete Task 2 in Python, as shown in Figure 5.5. We also noticed that Doppler references an external PyPI webpage (PyPI 2024) in their “Development/Editors” documentation which also shows an example of how to complete Task 2 in Python. Infisical throughout the time of our full study did not include such content in neither their public help documentation nor web UI. We decided to continue our experiments as planned and take note if these additions to the Doppler help documentation and HCP Vault Secrets web dashboard had

3 Read your secret

```
hcp vault-secrets secrets open {desired secret}
```

You may also inject secrets into your app as environment variables by passing a command as string, as shown below for an app using python.

```
hcp vault-secrets run -- python3 my_app.py
```

Figure 5.5: Additional part of HashiCorp vault secrets documentation in their web dashboard that reflects the required CLI command for Task 2.

any affect on the participants completing the tasks. We viewed this phenomenon as a natural experiment. Meaning, we observed differences in how participants performed on tasks using tool documentation when naturally presented with different content specificity.

Data Analysis

We recorded whether participants completed each task and their completion times for each task. We also recorded two specific actions from participants that indicate deviations from the expected or “recommended” path a user would take to complete these tasks in a real setting. The expected path we refer to is that participants complete a task using the SMT and steps recommended by the SMT tool documentation to complete each task within the time constraint. The two actions we recorded which represent a deviation from the expected path are usage of secondary sources and attempts at workaround methods.

The usage of secondary sources means that for a given task, a participant searched for and accessed a source outside of the tool documentation provided by the SMT in order to complete a task. By “workaround methods”, we refer to trying commands or actions that were not the recommended method suggested by the SMT tool documentation to complete a task (e.g., exporting tokens to access secrets or using the injection command to print the plaintext version of a secret in the CLI in Task 1). These actions allowed us to ask follow-up

questions in our post-task interviews to see why participants deviated from the expected path in order to complete a task. We present these findings in detail in Section 5.3.1.

5.2.3 Post-Task Interviews

Interview Guide Development

We interviewed participants after each task to get in-depth responses about their experiences during each task. We focused our questions on the usage of the SMT, tool documentation, and usage of secondary sources if they accessed them. To develop questions addressing this focus, we first referred to prior work on software documentation (Parnin et al. 2012; Acar et al. 2017b) and tool observation studies (Robillard 2009; Acar et al. 2017a). Prior work has highlighted content challenges such as insufficient code snippets, explanatory text, or lack of references for a wide range of use cases. Therefore, we used known related challenges from prior work as well as our research questions to draft several high-level categories of questions. We then piloted our interview guide with our observational study design with three PhD research students in our research lab to refine our interview guide and study design. Two of the PhD students pursue research in software engineering and software supply chain security, while the third PhD student pursues research in usable security and privacy. We updated our interview guide based on feedback from pilots. The complete interview guide is available in the Appendix Section C.1.

Interview Structure

Warm-up and wrap-up questions are asked before and after both tasks, respectively. The following high-level categories here represent questions we asked participants:

Warm-up Questions: Before Task 1, we asked a set of warm-up questions to learn about their academic and/or professional experience with secret management, as well as the types of secrets and tools they have used.

Overall Experience: The first question we asked after participants finish a task is to describe their overall experience with the task.

Negative Emotions: We asked participants if they felt any negative emotions when performing each task. Negative emotions can include annoyance, confusion, fatigue, frustration, or any form of disapproval/dissatisfaction.

Documentation Content: We specifically asked participants about challenges they experi-

enced using the content of the tool documentation itself to complete each task. Content includes (but is not strictly limited to) code snippets, code blocks, CLI commands, coding or web UI examples, explanatory text, images, or videos.

Documentation Structure and Formatting: We specifically asked participants about challenges they experienced with the structure and formatting of the tool documentation. Structure includes (but is not strictly limited to) section headers, bullet points, tabs, or toolbars. Formatting includes (but is not strictly limited to) text, pictures, or videos used in the advice.

Secondary Source Usage: We then asked participants about their usage of secondary sources and if those sources were more challenging to use or more helpful than the official tool documentation.

Wrap-up Questions: We wrap up the study by asking participants if they would consider using the SMT in the future, what their general preferences are for receiving information from sources, and for any last comments or questions about the study.

Analysis and Coding

Interview transcripts were analyzed through qualitative deductive and inductive coding. We first developed an initial codebook based on our interview and research questions. Using this initial codebook, two of the researchers met to code a subset of the transcripts and compare their coding results. After the subset coding, the two researchers then split coding evenly (13 and 14 transcripts coded respectively) and met weekly to discuss coding disagreements and revise the codebook to accurately reflect our research interests. The two researchers also double coded six of the transcripts and checked inter-coder agreement weekly McDonald et al. (2019b) to ensure we accomplished high agreement when independently coding transcripts (Krippendorff's Alpha > 0.75 for both researchers during independent coding Freelon (a,b)). We continued to iteratively code the interview transcripts until no new codes emerged and all transcripts were coded. The final codebook is in our Appendix Section C.2.

5.2.4 Data Protection/Ethical Standards

We took multiple steps to protect participants' privacy and data, while also ensuring all participants were treated ethically and with respect. Before participants performed our

study, we described our research goals to them and obtained consent from all participants. Participants were also informed on how their information would be protected on our consent form. During the study, we briefed participants on their rights during the study, including the right to skip any tasks and questions, and the right to withdraw without loss of benefits. We obtained video recordings and journal notes from each session, as well as written transcripts from audio recordings and performed both quantitative and qualitative analyses on our data. Written transcripts and journal notes were de-identified by replacing personally identifiable information (participant names, organizational names) with pseudonyms such as P01. We also did not request identifying, confidential, or private information about our participants or their prior experience in our study. We used end-to-end encrypted tools in all of our study communications and data storage components. Our study protocol was approved by our University's Institutional Review Board (IRB).

Further, we ensured that none of the CS Master's students held any prior relationships (academic or professional) or interactions with the researcher leading the in-person observational tasks and interviews. We also informed participants they would only receive monetary compensation, and their involvement would not result in any academic credit compensation or hold any bearing on their academic status.

5.2.5 Limitations

Participants were placed under two constraints during this study. The first being a 30 minute time constraint to complete each task. Such a time constraint may have influenced participants to overlook or skip steps that they normally would take in order to finish the task. Also, emotions of stress or fear of incompleteness may also have been a factor for participants during this study. We informed participants that their compensation is not dependent on performance, but rather based on the amount of time they take to complete tasks. We also allowed participants breaks at any time, allowed questions in which we pause the time, or allowed them to skip a task if they chose to do so. The second constraint is that participants only used one of the three SMTs during their tasks. We designed our study this way because we did not want to overburden or confuse participants with too many tools or tasks, and also to avoid learning effects.

5.3 Results

This section presents the findings of the challenges and experiences participants faced when using a SMT to perform two secret management related tasks. In Section 5.3.1, we report findings for participant task completion, completion times, usage of secondary sources, and attempts at workaround methods for each task. In Section 5.3.1, we present noticeable differences for participant performance for tasks across the three SMTs. The remaining sub-sections report in-depth qualitative findings from post-task interviews with participants who reported their challenges and experiences using SMTs to complete our secret management tasks. Our qualitative findings aim to explore the context for why developers widely report setup and learning constraints for SMTs when adopting them (Krause et al. 2023; Basak et al. 2023b). In Section 5.3.2, we report how participants described their overall experience with each task. In Section 5.3.3, we report specific challenges with the tool documentation that contributed to reported task challenges and deviations from the expected methods as provided from the tool documentation. In Section 5.3.4, we explain responses participants gave for why they accessed secondary sources, and also explain how helpful participants reported that access to secondary sources was for their tasks. Lastly, in Section 5.3.5 we report participant insights and responses towards the general usability of the SMTs they used.

5.3.1 Participant Performances

Tables 5.2 and 5.3 show the performance statistics for participants. Overall, we observe noticeable differences in participant performance between tasks. 21 participants completed Task 1 within 30 minutes, whereas only 18 participants completed Task 2 within 30 minutes. We also notice that participants on average completed Task 1 in 11 minutes and 8 seconds, whereas participants on average completed Task 2 in 16 minutes and 18 seconds. More importantly, we noticed that only eight participants accessed a secondary source or tried a workaround method in Task 1, while 16 participants accessed a secondary source or tried a workaround method in Task 2. 12 participants in Task 2 accessed a secondary source, and only two participants accessed a secondary sources in Task 1. Twice as many participants tried a workaround method in Task 2 (14) as they did in Task 1 (7).

Our findings indicate that the number of participants that deviated from the expected path, as described in Section 5.2.2, doubled in count from Task 1 to Task 2. More than

Table 5.2: Participants' Performance in Task 1.

	Completed	Time	Secondary Source	W.A Method	Either Or
P01	✓	8:57			
P02	✓	17:03			
P03	✓	7:10			
P04	✓	19:45	✓		✓
P05	✓	7:56		✓	✓
P06	✓	11:52		✓	✓
P07	✓	5:40			
P08	✓	9:43		✓	✓
P09	✓	5:35			
P10	✓	10:39		✓	✓
P11	✓	14:03			
P12	✓	3:34			
P13	✓	12:17			
P14	✓	23:42	✓	✓	✓
P15	✓	6:44		✓	✓
P16	✓	1:52			
P17	✓	23:28		✓	✓
P18	✓	22:07			
P19	✓	10:33			
P20	✓	8:31			
P21	✓	2:41			
Overall	21	11:08 [†]	2	7	8

[†] Average

two-thirds of participants struggled to solely stick with the provided tool documentation and the expected methods for each task. We emphasize this as a focus in our findings since Task 1 requires several more individual steps than Task 2. Task 1 requires participants to remove `CompanySecretToken` from `SecureSecret.py`, store `CompanySecretToken` in the SMT web UI `CompanySecretStorage`, login and initialize into the SMT CLI, and finally print the value of `CompanySecretToken` from the command line. Task 2 on the other hand only requires the participant to input one command in the SMT CLI which performs the injection run process as shown in Figure 5.4. It is unclear what specifically contributed to different paths participants took for each task by only looking at our data from Tables 5.2

Table 5.3: Participants' Performance in Task 2.

	Completed	Time	Secondary Source	W.A Method	Either Or
P01		>30:00	✓	✓	✓
P02		>30:00	✓	✓	✓
P03	✓	12:13	✓		✓
P04	✓	5:18		✓	✓
P05	✓	4:04			
P06	✓	13:28	✓	✓	✓
P07	✓	16:23	✓		✓
P08	✓	7:50		✓	✓
P09	✓	7:15	✓	✓	✓
P10	✓	1:04			
P11	✓	21:03	✓	✓	✓
P12	✓	4:37			
P13	✓	13:20	✓	✓	✓
P14		>30:00		✓	✓
P15	✓	23:23	✓	✓	✓
P16	✓	16:44		✓	✓
P17	✓	23:10	✓	✓	✓
P18	✓	29:05	✓	✓	✓
P19	✓	22:51			
P20	✓	2:00			
P21	✓	28:44	✓	✓	✓
Overall	18	16:18 [†]	12	14	16

[†] Average

and 5.3. Therefore, we use Sections 5.3.2, 5.3.3, 5.3.4, and 5.3.5 to find explanations for why participants experienced increases in measured difficulty and deviations from solely relying on tool documentation from the SMTs.

Task Performance Comparing SMTs

Here, we briefly present differences in participant performance statistics we noticed between the three SMTs for each task.

Task 1: First, we examine how SMTs vary in their time-to-completion. Doppler (med. 11.9 minutes, $\sigma = 6.41$) and Infisical (med. 10.7 minutes, $\sigma = 6.84$) have task dis-

tributions with a center approximately double the median completion time of HCP Vault Secrets (med. 5.58 minutes, $\sigma = 4.25$). None of the seven HCP Vault Secrets participants accessed secondary sources or tried workaround methods. Whereas Infisical had the most people trying secondary sources and workaround methods. Because we only have seven participants in each group, statistical power is quite limited, and hypothesis tests should not be considered definitive. Nevertheless, we performed a Kruskal-Wallis test out of curiosity. We found a statistically significant difference in distributions ($H = 6.50, P = 0.39$) with a “large” effect size ($\eta^2 = 0.250$).

Task 2: We repeat the same analysis for Task 2. Unlike Task 1, we had three participants who failed to complete the task. Two of those participants used Infisical, which was the same SMT with the most participants not completing Task 2. Infisical also had the lowest number of participants who accessed secondary sources. We arbitrarily assigned a value of 30 minutes time for these participants, though we find using a higher value did not change the statistical conclusions. Doppler (med. 22.9 minutes, $\sigma = 7.53$), Infisical (med. 5.3 minutes, $\sigma = 12.85$) and HCP Vault (med. 16.74 minutes, $\sigma = 9.95$) vary widely in median but largely have overlapping distributions. We also performed a Kruskal-Wallis test on these times, finding no statistically significant difference in the distributions ($H = 1.41, P = 0.495$). Again, with low sample sizes these findings should be considered provisional.

5.3.2 Overall Experience and Challenges

Overall, participants experienced a neutral or positive experience when performing Task 1, but they experienced a negative experience in Task 2. We did not observe any significant responses for negative emotions experienced during Task 1 outside of sparse mentions of confusion or annoyance towards either CLI command inputs or explanations in the tool documentation.

Task 1: 13 participants agreed with P03 who said Task 1 was “*pretty simple and straightforward*”. Participants generally understood the CLI commands provided from the tool documentation that were intended for Task 1. P04 stated - “*I think it is very easy to set up, and provides good APIs for managing secrets...*”. Participants largely felt that the tool documentation for this task was well written and offered helpful information such as quick start up guides as P11 mentions:

P11 *Now, coming to the particular task. I felt like the documentation was very well written, and it has a quick lessons for people to learn and adapt...*

Task 2: On the flip side, participants reported the bulk of negative emotions and experiences came from Task 2. 15 participants reported a negative experience or emotion during Task 2, with several participants specifically saying Task 2 was more difficult than Task 1. P13 stated the following:

P13 *Yeah this task was... a little bit uh... difficult than the first task. Like it made me search the documentation very well.*

Participants most commonly felt confused or frustrated when trying to learn how to perform injection while looking at tool documentation. Both P09 and P15 mentioned they knew they had to inject `CompanySecretToken` via command line, but could not figure out how to do it by reading their tool documentation for injecting secrets. P15 stated: *“The documentation doesn’t show you exactly how inject commands into the program”*. Nine participants reported that even finding helpful information for secret CLI injection in their tool documentation was a challenge. P08 mentioned that relevant documentation for the injection run command is *“buried inside the docs”*. Even when participants left the tool documentation to search secondary sources, few found helpful sources. P03 mentioned the Google pages with steps were not helpful to them. Challenges with using available tool documentation prompted participants to try workaround methods or access secondary sources to complete their task, primarily Task 2. Resorting to workaround methods, secondary sources, and going back and forth with the tool documentation provoked negative emotions, such as confusion and frustration, as reported by participants. P16 mentioned they *“found it frustrating that the documentation was not properly documented”*. P01 expounded on their frustration during Task 2 with the following:

P21 *I would say after 20 minutes into the task 2, I was feeling a little frustrated... after we reached 25, 26 minutes, ... my thoughts were like, okay, now that I have taken so much time for a simple task, maybe the solution is very simple.*

5.3.3 Tool Documentation Challenges

In this section, we specifically focus on challenges and experiences directly reported from using SMT tool documentation and web UI to complete each task.

Documentation Content

Tool documentation was widely reported as providing enough information and being generally helpful by participants during Task 1. 13 participants even cited specific documentation they found helpful. For example, both P04 and P10 mentioned that the Infisical CLI quickstart guide was helpful for managing secrets and gave basic steps to do so. P03 reported that HCP Vault Secrets was “*easy to navigate*” and appreciated the usage of images in documentation. P18 and P19 appreciated CLI command descriptions in Doppler’s CLI Guide.

There were a few instances when participants mentioned a lack of clarity over CLI descriptions and usage during Task 1. P09, P11, and P19 expressed confusion on applying CLI secrets commands when command line arguments were not explicitly defined on the same page in the documentation. Participants reported confusion over CLI usage when they felt the content did not sufficiently cover the relationship between CLI commands and secret management processes. Both P04 and P19 felt the content they used did not fully address immediate actions required to complete the initialization and login process for using the CLI.

Participants who attempted workaround methods for Task 1 mostly tried CLI commands to export secrets and also the injection run command which was needed for Task 2. None of the participants were aware they were going to inject secrets before Task 2, so some of them used the injection run command to perform injection to complete Task 1. P15 mentioned the following as to why they used the injection run command for Task 1:

P15 *So my task was basically to output the secret, right? So the first thing that I did was like doppler run...*

For Task 2, participants largely reported tool documentation as being much less helpful and consequently reporting Task 2 as more challenging than Task 1. 17 participants reported that content for the injection run command contributed to challenges they experienced when completing Task 2. Participants mentioned the following problems: lack of sufficient information for relevant command line arguments, lack of a relevant set of use cases in examples provided, and inconsistent documentation that covers secret injection.

Command Line Arguments: First, several participants mentioned that command line arguments were introduced in the tool documentation for the injection run command. However, they were confused reading because they felt there was a lack of explicit information on

how to apply them and how they related to the injection run command. P09 and P16 both referred to injection run command documentation from HCP Vault Secrets CLI page. They did not understand the `duration` flag shown at the end of the command in the documentation. P09 also stated the documentation did not mention what the standard double dash option argument was for. P16 tried using the `duration` flag in their injection run command attempts but could not figure out how to make that flag work with the command. P16 also explained how they say another flag mentioned in the documentation that was also not easy to understand to them:

P16 *And the first thing, if you see, they have mentioned `app`, `my-app`, the application you want to pull all secrets from. But you don't understand its purpose or it doesn't say what happens if you don't do it. I had to try it myself.*

Example Use Cases: Second, six participants explicitly stated examples provided from the tool documentation were a challenge when figuring out how to apply the injection run command to the study Python file, `SecureSecret.py`. Before our full study, injection run command examples shown in all of the SMTs tool documentation did not explicitly present a Python example, but other languages and frameworks such as Flask, Golang, NPM, and command line functions like `echo`. Participants wanted more clarification from tool documentation on how languages or frameworks used in the examples are simply just examples and not required to run the command itself. P17 explained how they were initially confused with the explicit injection content in Doppler's Secret Access CLI Guide since it primarily used an example with Node Package Manager (NPM) without any other examples present:

P17 *This `npm start` is just a command example... They should have mentioned that `npm start` is just an example you can type in your command instead of `start`. That got me a little confused. Otherwise, the task was easier but it took me some time to figure out the exact solution.*

Inconsistent Command Presentations: Third, several participants mentioned they observed discrepancies in relevant command snippets and arguments used in content describing secret injection across different documentation pages. Multiple participants that used either Inifisical or HCP Vault secrets reported that both tool's quick start guides and their CLI documentation pages differed in both the examples and arguments present for their injection run command descriptions. P05 mentioned that while Inifisical's injection run

command page provides a bare example without flags or arguments, the Infisical quickstart guide only provides examples with sample flags and arguments. P05 later explained that led them to try the injection command with incorrect flags and arguments because they did not know beforehand those were not necessary, thus received errors at first. Participants that used HCP Vault Secrets faced this challenge even with the existence of added dashboard documentation that provides a specific Python3 example for the injection run command, as shown in Figure 5.5. For participants who noticed the added dashboard content, they mentioned that the inclusion of the Python injection run command example helped them complete the task. However, they clarified it was not intuitive to expect secret injection documentation within the web dashboard or outside of the public tool documentation. P11 noticed the Python example in the HCP Vault Secrets dashboard and stated that “*maintaining two different pages and they have two different contents, obviously confuses people*”. P03 echoed a similar response:

P03 *I would say the first one was easier because the links to the documentation were available... In my mind, I was assuming that there will be similar documentation available somewhere. And it did not strike me that I can just take a look at the portal... I was expecting to find a similar web page where the information that is provided for the first task would be provided for the second task. (P03)*

We believe insufficient information, a lack of relevant use cases in examples, and inconsistent documentation contributed to participants deviating from strictly following tool documentation. From our interview responses, participants reported struggling with understanding how to perform secret injection in Python when reading tool documentation command line arguments and examples. Inconsistent presentations of injection run commands also confused participants as to how to perform secret injection in Python.

Documentation Structure and Formatting

Here we report responses from participants towards the structure and formatting of the tool documentation. We also briefly discuss specific navigation difficulties participants experienced with the search functionality in the tool documentation.

Structure: The structure of the tool documentation received generally positive feedback. Participants appreciated the overall clarity in structure, including well-defined headers and organized sections like P18 mentioned, “*everything is labeled nicely*”. Both P04 and P20 felt

the web UI for the tool documentation was clear and minimalist in design. P04 specifically appreciated when titles, headers, and tabs were used to separate topics.

Content Format Preferences: Regarding content format, participants had mixed preferences. Some participants preferred text for its searchability and ease of reference. When asked the reason for preferring text, P09 explained they “*prefer reading text over videos*” as it allows them to search for keywords and scan information faster. Other participants found code examples and videos beneficial for visual learning. For coding issues, P21 mentioned, “*text with code blocks is very helpful*”. P17 explains their preference for videos here:

P17 *First reference would be video because if I want to get help for something then it would be better for me to see that the other person is doing the same thing and explaining it in a video.*

Few participants mentioned their content preference depended on the complexity of the task at hand. P01 stated, “*minor problem may be a text, a bigger problem somewhere in between text and video*”. Overall, participants held mixed preferences regarding format, suggesting that offering mixed formats could better support varied user needs.

Navigation Difficulties: During both tasks, participants reported specific navigation difficulties for the tool documentation. P04 felt that the “*search functionality could be improved*” in the documentation, as they struggled to find specific terms or commands, and mentioned that the “*ease of search was missing*.” Participants found it most challenging to use the search functionality from the tool documentation to find sufficient information for secret injection, as P04 mentioned:

P04 *... when I typed injection, it did not give me the desired response. So I think it is good that they are returning an LLM response, but I would also appreciate if they would have returned a link to where I could find what I actually wanted. They're basically giving me a ChatGPT kind of response, but I would also appreciate the source of that thing.*

As navigation difficulties increased for participants during Task 2, their time and effort also increased as they needed to perform more troubleshooting and exploration.

5.3.4 Secondary Sources

Here, we focus on participants responses towards why they accessed secondary sources and how helpful they reportedly found secondary sources. We do not analyze the secondary

sources in depth themselves (e.g., content, structure, formatting, or source of origin). We only provide some examples and briefly discuss relevant content from examples participants mentioned where relevant.

Only two participants during Task 1 accessed secondary sources online for further assistance, as shown in Table 5.2. One of those participants, P14, mentioned they accessed a secondary source to reduce confusion for the injection run command by finding more examples. As we previously mentioned in Section 5.3.3, a few participants from Task 1 attempted the injection run command intended for Task 2 because they thought that command would produce the intended output required for Task 1. This led P14 to seek secondary sources on how to use the injection run command during Task 1. We explain implications behind how this can lead to complications broadly for users adopting new software technology through tool documentation in Section 5.4.2.

Seeking further assistance on how to perform secret injection in Python became the common reason for participants accessing secondary sources during Task 2. As shown in Table 5.3, twelve participants searched for and accessed secondary sources outside the tool documentation during Task 2. Ten of those participants explained they needed information which was not addressed in the tool documentation. P06 stated the tool documentation was not necessarily confusing, but they needed *“maybe a little more description”*. P21 felt the examples provided were not elaborate for completing Task 2. This reasoning led participants to search the web for secondary perspectives on how to inject secrets with their SMT. P18 described this process:

P18 *I scrolled quickly through the main documentation and I was stuck. So it's like whenever you are stuck at something, you need some other perspective.*

Finding solutions from others online motivated participants in our study to use sites like StackOverflow. P15 felt that while the Doppler CLI Guide was *“really good to get started”*, StackOverflow is better for finding the *“specific piece of code you are actually looking for”*. P21 mentioned they appreciate added background and debugging advice from users on StackOverflow. Participants who searched for secondary sources often indicated those sources helped them more than the tool documentation when they received that added bit of information or example usage which they felt was missing from the tool documentation, specifically for Task 2. P06 googled broadly for information for running Doppler with Python files and mentioned that a Dev Community blog post (Community 2024) gave better descriptions for running the injection run command with Doppler for Python files.

More often than not however, participants would either try multiple search queries and multiple distinct secondary sources before either finding one that helped or eventually going back to the tool documentation for Task 2. P03 even mentioned that when they did try to Google for secondary sources, the results returned “*were totally irrelevant.*” P15, while searching for sources, clicked on a blog post by Medium Corp. (Medium 2024) which they believed would give relevant advice for secret injection. However, P15 later clarified that the blog “*was very complex and covered kubernetes. It didn’t actually tell me what I wanted.*”

As mentioned in Section 5.2.2, Doppler provides a hyperlink within the Development/Editors Section (Doppler 2024) for Python which directs users to a external doppler-env-package page (PyPI 2024) that shows an injection run command example in Python. While three of the Doppler participants specifically mentioned that the Python example for secret injection from the doppler-env-package page helped, finding the page was a challenge. P15 mentioned that doppler-env-package page (PyPI 2024) “*wasn’t really anywhere in the documentation.*” P18 on the other hand noticed the hyperlink for the doppler-env-package page, however the surrounding headers and descriptions mentioned configuration and installation steps. Therefore, P18 early only thought any hyperlinks within that section were “*only about installation.*” They did not reach the doppler-env-package page until Googling for more documentation and being directed to it.

The motivation from participants to acquire secondary sources increased noticeably when the tool documentation provided by the SMTs were perceived to not be sufficient enough in their explanations or examples, or provided inconsistent information. Even with the direct Python examples of injection run commands from HCP Vault Secrets web dashboard and Doppler’s external doppler-env-package page, participants experienced difficulty in finding that information. HCP Vault Secrets participants did not expect relevant content for their tasks to be in their web dashboard, and Doppler participants could not find the doppler-env-package hyperlink which is mentioned in a section not specifically related to secret management. Other secondary sources were only helpful if they gave descriptions or code snippets participants felt that the tool documentation did not provide. However, most of the secondary sources accessed by participants were not much more helpful than the tool documentation, leading participants back to the tool documentation to finish Task 2.

5.3.5 Tool Usability Feedback

Participants highlighted both positive aspects and suggestions for the SMTs from the study. The copy to clipboard functionality was appreciated by P19, who mentioned they can just copy to the clipboard and paste it in the “*terminal directly*” along with the ease of creating tokens as mentioned by P16,

P16 *I liked how easy it was to create a token, to see the token and it's like if there's a company that's using this application, new developers are going to join and may use it more easily. There's no complexity.*

P21 also mentioned that the web dashboard was more helpful while working on the task, with its “*UI is particularly helpful*” to see all the secrets, and the dashboard allowing to see all the applications at once. Many saw the SMT as a “*good option for managing secrets*” with several others expressing interest in using it for future projects and exploring its integration capabilities. Participants also compared the tool to others, some felt it is a strong alternative to Git Lab secrets, while others mentioned preferring established cloud options like AWS for its documentation and familiarity like P15 stated,

P15 *I'm more of an AWS guy and it already has something like KMS. Like if you use AWS, KWS, EC2, you really don't have to do anything else, it is all in one place. I don't think I'd want to use Doppler. Also, the documentation for KMS is really nice, people have been using it for a while.*

In terms of personal versus professional usage, participants like P01 generally felt the tool was more suited for “*company or team-based projects rather than personal use*”. P09 mentioned that the “*pricing structure of the tool*” is one of the deciding factors for adoption. Some participants like P04 also mentioned that the tool was easy to set up and had clear documentation and an intuitive user interface. P13 felt that the dashboard and command-line interface further improved the overall usability by stating,

P13 *I think the tool was very helpful in managing the secrets. I like the way we can interact with the dashboard like they have created a dashboard that then we can interact via the visual interface and we can import the secrets as well if we don't want to use the command line. Whereas people who love the command line, have a dedicated command line as well. So I like that idea pretty much.*

Participants also pointed to various features that supported their needs where P16 appreciated the *“availability of API, CLI and customization”*. Whereas, P08 also pointed out that for projects, relying on a tool like Doppler *“could create dependency and difficulties in migration”*. P01 faced login issues, mentioning they *“didn’t know how the login works”*. P15 also highlighted problems with the save feature, stating they *“didn’t really like it”* and P04 also pointed out issues with handling special characters, stating it *could improve on string escape characters”*.

5.4 Discussion

We discuss how our findings provide context for why new users may report challenging experiences when first setting up and using SMTs to manage secrets. We use both our participant performance and interview findings to complement each other. We conclude with providing recommendations for SMT providers when designing SMT tool documentation and interfaces.

5.4.1 RQ1: New Users and SMT Usage

Participants joined our study with a baseline of prior experience in secret management and related programming concepts from our eligibility criteria. Some held experience using SMTs not from this study while others used a SMT for the first time during this study. Our findings support prior work towards identifying how developer experiences using new tools can be drastically impacted by the quality of available resources present (Gorski et al. 2018; Basak et al. 2022; Krause et al. 2023; Acar et al. 2016). The ability for new users in our study to effectively store, access, and inject a secret with a SMT in one hour changed drastically between each task. Even though participants completed Task 1 with relative ease, Task 2 is the process which helps ensure users can securely run local applications without hardcoding secrets. Both tasks were designed to represent fundamental processes for SMTs that allow users to securely manage secrets, while being short enough to complete back to back in a one hour laboratory setting. If participants in our laboratory setting struggled to effectively perform both tasks, then it is likely new users in a real world setting experience similar challenges with even less guidance on how to effectively use SMTs to securely manage secrets, which demonstrates a contribution to adoption burdens reported by developers in prior work Krause et al. (2023); Basak et al. (2023b).

5.4.2 RQ2: Factors that Impact SMT Usage

We discuss positive and negative aspects about the experiences reported by participants using SMTs and tool documentation to manage secrets.

Positive Factors and Experiences

Participants provided multiple positive statements about the interfaces and tool documentation they used with the SMT, mostly while performing Task 1. Similar to prior efforts (Acar et al. 2017a; Smith et al. 2020; Indela et al. 2016; Nassif et al. 2021; Lethbridge et al. 2003), participants in our study largely appreciated tool documentation and interfaces that simplified their process of finding relevant information and provided just enough information to complete each task. Participants felt that the tool documentation provided enough information to complete Task 1 and a web interface that was intuitive enough for them to find all of the information they needed. The inclusion of quick starter guides, copy-to-clipboard options for command snippets, and structural features like well-defined headers and organized sections were common aspects participants felt made their experience easier. We provide further insight from prior work in directly observing that as participants felt more positive about tool documentation and interfaces, they completed their task in fewer time with less challenges along the way. Most importantly, we saw that participants were less likely to deviate the expected path while using SMTs to manage secrets when they reported positively about tool documentation and interfaces. Thus, decreasing the risks of traversing through irrelevant information, implementing insecure methods, or an inability entirely to learn how to use new tools.

Negative Factors and Experiences

Our findings support related efforts by continuing to highlight tool documentation challenges such as insufficient coding examples (Aghajani et al. 2020, 2019; Nassif et al. 2022; Acar et al. 2017b,a; Subramanian et al. 2014) and incorrect or ambiguous explanations (Uddin and Robillard 2015; Wen et al. 2019; Middleton et al. 2020; Chen and Huang 2009; Acar et al. 2017b; Sohan et al. 2017; Treude and Robillard 2016). However, our work extends prior efforts by exploring challenges participants report with using SMT-specific tool documentation covering CLI commands. New users already face a learning curve when first adopting SMTs (Krause et al. 2023; Basak et al. 2023b). If new users encounter tool doc-

umentation with insufficient descriptions for relevant CLI command line arguments, or without a broad range of example use cases for CLI commands, they are less likely to know how to effectively use the SMT CLI functionality. Therefore, new users are also less likely to understand how to securely manage their secrets with SMTs when reading provided tool documentation. We discuss recommendations that SMT providers should consider when writing tool documentation to reduce participant confusion when learning SMT CLI commands in Section 5.4.4.

Our work also provides more context towards the experiences new users have when searching secondary sources for technical assistance as highlighted in prior work (Arya et al. 2023; Baltes et al. 2020; Robinson et al. 2022; Storey et al. 2024; Acar et al. 2016; Parnin et al. 2012). Most relatedly, Acar et al. (Acar et al. 2016) observed that Android code written by developers produced different levels of security and functionality depending on the information source the developer used for assistance. In our study, participants complained that information from secondary sources was either no different than what the tool documentation provided, overly complex for their task, or completely irrelevant. In a real world setting, new users learning a tool search for information sources and need to determine if the source they are looking at is proposing methods that will help them achieve the expected outcome they require. The extent of harm for applying incorrect, or worse insecure, methods increases if new users feel inclined to spend more time exploring secondary sources after unsuccessfully trying official tool documentation. Therefore, making it more difficult for new users to effectively use or adopt new tools. We believe our findings for participants using SMT tool documentation and interfaces add context to adoption burdens reported by developers (Basak et al. 2023b; Krause et al. 2023). Specifically, if SMT tool documentation is challenging to use, and secondary sources discussing SMTs only add more confusion, then developers may be less willing to further adopt SMTs. Developers interested in adopting SMTs may have challenges using SMT tool documentation to manage their secrets or navigating different information sources to find relevant solutions for their use cases.

5.4.3 RQ3: Comparison of Different SMTs

We performed statistical analyses to see if we could identify any differences in participant performances across the three SMTs. While we do not claim definitive statistical significance, we found median completion times across SMTs varied widely, but largely overlapped in

range. For example, in Tables 5.2 and 5.3, three and two participants completed Task 1 and 2 in under four minutes, respectively. On the flip side, three and nine participants took at least 20 minutes to complete Task 1 and 2, respectively. Thus, we see the possibility for participants to effectively use SMTs to secure secrets quickly, but we also see many other participants look at the same documentation and take considerably longer. Therefore, we reason the SMTs in our study were functionally indistinguishable compared to each other when participants used them to manage secrets. This further supports our findings that the quality of SMT tool documentation was more impactful towards SMT usage success.

5.4.4 Recommendations

Here, we offer recommendations to SMT providers regarding tool functionality and documentation.

Dedicated Sections for Technical Terms: Many participants recommended including dedicated sections in the documentation to explain technical jargon, such as command flags, subcommands, and arguments. These sections could provide clear definitions, examples, and a glossary to help users, especially beginners, navigate the documentation more effectively.

Visual Aspects for Demonstration: Additionally, participants also suggested including visual aspects in the documentation like flow diagrams to demonstrate how commands interact with values, arguments, and output. These diagrams could provide a clear logical flow for the key tasks in the SMTs like logging in, initializing processes, and accessing secret projects. While resolving issues, users could also benefit from having debug examples, code snippets, and guidance.

Structural Improvements: Another recommendation is regarding the structuring of CLI documentation with a base example at the top, followed by examples with subcommands, flags, and arguments. This approach keeps all relevant information on one page, reducing the need for users to search elsewhere, and improving the overall user experience by minimizing lookup time. An effective structure and formatting with clear headings, a logical flow, and tables to represent a list of commands, subcommands, and argument explanations would help users quickly access and understand the information. Also, adding hyperlinks or resources in the documentation to guide users on installing necessary programming environments or dependencies would help beginners to quickly set up and follow the rest of the instructions to complete their tasks.

Clear Placeholders in Commands: Participants reported confusion over which CLI argument parameters were necessary to replace or use verbatim. Documentation authors should use obvious placeholders (e.g., `your_project_name`) to help users avoid confusion.

User-friendly Features: Providers are also recommended to continue to include user friendly features like copy to clipboard for ease of use and also consider adding search functionality and quick navigation links to improve the documentation usability and help users complete their tasks more efficiently.

5.5 Conclusion

In our exploratory mixed methods laboratory study into the usage of SMTs, we observed how differences in available tool documentation content and help resources impact the ability to use SMTs to perform secret management tasks. We identified specific challenges relating to tool documentation content, structure, and formatting that participants reported impacted their ability to manage secrets with an SMT in our study. Challenges participants encountered with using tool documentation prompted them to deviate towards accessing secondary sources and attempting workaround methods while cycling through tool documentation and experiencing added difficulty. We make recommendations for both SMT vendors and tool documentation writers to improve the ecosystem of guidance for new SMT users.

CHAPTER

6

CONCLUSION AND FUTURE WORK

In this dissertation, we investigated challenges that affect the quality and usability of online security advice, as well as the usability for developer secret management tools. In Chapter 3, we investigated the coverage of account remediation from 57 popular U.S.-based web services. We introduced a systematic model to define account remediation and used that model to qualitatively code web pages posted from web services in our study that discussed end-user advice for account compromise remediation. We found that 61% of popular U.S.-based web services in our study failed to provide advice for all five phases for account remediation. In Chapter 4, we interviewed 21 authors of general security advice to learn what are the key processes, decision making, and challenges advice writers face when writing general security advice. We find that advice writers struggle to gather and prioritize the information necessary to write advice content, thus having a trickling effect into the processes and key decisions made in advice writing. Our contributions help provide insight into the context and reasons for a lack of consensus on security advice, as well as for why advice content experiences variable amount of coverage. In Chapter 5, we performed a mixed-methods study with 21 CS Master's students with prior secret management experience to see how effectively they could use SMTs to perform two secret management tasks:

Secret storage and access, then secret injection. We identified specific limitations within SMT tool documentation that negatively impacted SMT usage and hindered participants performances to complete our two secrets managements tasks. We further observed that when participants encountered tool documentation limitations, they experienced added difficulty in troubleshooting or exploring solutions from either the tool documentation or secondary sources.

Our work was motivated by several known challenges about security advice that was highlighted from prior research efforts. We sought to further investigate security advice challenges to learn why everyone struggles with security advice. While Chapter 4 is our second work, it investigated the source of security advice creation and how an insufficient foundation for key decisions and processes can affect the coverage of numerous security advice topics. As we see in Chapters 3 and 5, we observed differences in the coverage of important security advice topics that were relevant to online account security and the security for developer software secrets, respectively. In Chapter 3, we use our findings to highlight the likelihood that end-users are not receiving sufficient information to remediate their compromised online accounts. However, we extend that takeaway by actually observing how differences in available tool documentation impacts the experiences for new developers who want to adopt SMTs, but cannot effectively use SMTs with available tool documentation provided online by the SMT.

Future research investigating security advice and developer secret management practices still remain. This dissertation motivates the following areas for further investigation:

6.1 Account Remediation Support

Future work should explore more usable or contextual guidance. Some of the steps in account remediation are technically complex to perform for users. Making the process of account remediation more usable and easier to follow will better aid users in remediating their accounts. For example, Facebook actually implements a chatbot-style wizard for guiding users through account remediation. It consists of easy to read diagrams that prompts users if they recognize information or settings on their account that is presented to them by the chatbot wizards. Future work could evaluate these approaches and explore ways of generalizing this approach to be usable for other types of web services beyond social media. Additionally, it is worth exploring to what extent a service could certify that an account has

been remediated, or what assurances could be provided to users that their accounts have become “safe.”

6.2 Secret Management Tools and Documentation

- **AI Tools as Information Sources:** While in this study participants were not allowed to use any AI tools, many frequently referred to options like ChatGPT and Claude AI for valuable resources on troubleshooting and coding support in their everyday work. Participant responses highlighted a shift from traditional resources, such as Google and StackOverflow, to AI tools. However, participants also described choosing tools based on task complexity and the nature of the problem. Future work can investigate the role of AI tools in developers’ tasks, particularly in comparison to traditional resources like official documentation or Stack Overflow. The study could include analyzing how AI tools influence developers’ task completion capabilities, problem-solving approach, and learning process as well as understanding any potential drawbacks of overly relying on AI tools. Based on our findings, many tools like SMTs can benefit from the integration of AI features.
- **SMT Documentation Format:** In the study many participants had diverse feedback on the format of advice, so for future research, we could compare the effectiveness of different documentation formats such as text, visual media, and mixed formats. We could analyse how these formats influence task completion, and satisfaction of users when working on different complex tasks. The findings could help propose a better format for the reachability of advice and documentation to users. Additionally, a follow-up study to examine the documentation of cloud-based or open-source SMTs.
- **SMT Integrations:** Exploring how SMTs integrate with other development tools like GitHub, Docker, and Kubernetes would give us more understanding and help identify potential challenges and improvements for both the documentation and tool interfaces of SMTs.
- **Security Implications for SMTs:** Although none of the participants in our study mentioned security concerns, it is important to explore how developers perceive and handle the security implications when using SMTs. Future work can analyze the user

awareness of the security features of SMTs, how to safeguard sensitive information, and potential knowledge gaps in the security practices of SMT users.

REFERENCES

- Abrams, L. (Online; accessed September 30, 2023). *New York Times source code stolen using exposed GitHub token*. <https://www.wiz.io/blog/38-terabytes-of-private-data-accidentally-exposed-by-microsoft-ai-researchers>.
- Acar, Y., Backes, M., Fahl, S., Garfinkel, S., Kim, D., Mazurek, M. L., and Stransky, C. (2017a). Comparing the usability of cryptographic apis. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 154–171. IEEE.
- Acar, Y., Backes, M., Fahl, S., Kim, D., Mazurek, M. L., and Stransky, C. (2016). You get where you're looking for: The impact of information sources on code security. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 289–305. IEEE.
- Acar, Y., Stransky, C., Wermke, D., Weir, C., Mazurek, M. L., and Fahl, S. (2017b). Developers need support, too: A survey of security advice for software developers. In *2017 IEEE Cybersecurity Development (SecDev)*, pages 22–26. IEEE.
- Aghajani, E., Nagy, C., Linares-Vásquez, M., Moreno, L., Bavota, G., Lanza, M., and Shepherd, D. C. (2020). Software documentation: the practitioners' perspective. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 590–601.
- Aghajani, E., Nagy, C., Vega-Márquez, O. L., Linares-Vásquez, M., Moreno, L., Bavota, G., and Lanza, M. (2019). Software documentation issues unveiled. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1199–1210. IEEE.
- Akhawe, D. and Felt, A. P. (2013a). Alice in warningland: A large-scale field study of browser security warning effectiveness. In *Presented as part of the 22nd USENIX Security Symposium*, pages 257–272.
- Akhawe, D. and Felt, A. P. (2013b). Alice in warningland: A large-scale field study of browser security warning effectiveness. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 257–272, Washington, D.C. USENIX Association.
- Arya, D. M., Guo, J. L., and Robillard, M. P. (2023). How programmers find online learning resources. *Empirical Software Engineering*, 28(2):23.
- Arya, D. M., Guo, J. L., and Robillard, M. P. (2024). Why people contribute software documentation. In *Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering*, pages 91–96.
- AWS (Online; accessed September 5, 2023). *AWS Key Management Service*. <https://aws.amazon.com/kms>.

- Bai, W., Namara, M., Qian, Y., Kelley, P. G., Mazurek, M. L., and Kim, D. (2016). An inconvenient trust: User attitudes toward security and usability tradeoffs for {Key-Directory} encryption systems. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 113–130.
- Baltes, S., Treude, C., and Robillard, M. P. (2020). Contextual documentation referencing on stack overflow. *IEEE Transactions on Software Engineering*, 48(1):135–149.
- Barbour, R. S. (2001). Checklists for improving rigour in qualitative research: a case of the tail wagging the dog? *British Medical Journal*, 322(7294):1115–1117.
- Basak, S. K., Neil, L., Reaves, B., and Williams, L. (2022). What are the practices for secret management in software artifacts? In *2022 IEEE Secure Development Conference (SecDev)*, pages 69–76. IEEE.
- Basak, S. K., Neil, L., Reaves, B., and Williams, L. (2023a). Secretbench: A dataset of software secrets. *arXiv preprint arXiv:2303.06729*.
- Basak, S. K., Neil, L., Reaves, B., and Williams, L. (2023b). What challenges do developers face about checked-in secrets in software artifacts? *arXiv preprint arXiv:2301.12377*.
- Beautement, A., Sasse, M. A., and Wonham, M. (2008). The compliance budget: managing security behaviour in organisations. In *Proceedings of the 2008 New Security Paradigms Workshop*, pages 47–58.
- Bonneau, J., Bursztein, E., Caron, I., Jackson, R., and Williamson, M. (2015). Secrets, lies, and account recovery: Lessons from the use of personal knowledge questions at google. In *Proceedings of the 24th International Conference on World Wide Web*, pages 141–150.
- Bonneau, J., Herley, C., Van Oorschot, P. C., and Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*.
- Boyd, M. J., Sullivan Jr, J. L., Chetty, M., and Ur, B. (2021). Understanding the security and privacy advice given to black lives matter protesters. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18.
- Bravo-Lillo, C., Cranor, L. F., Downs, J., and Komanduri, S. (2010). Bridging the gap in computer security warnings: A mental model approach. *IEEE Security & Privacy*, 9(2):18–26.
- Budiu, R. (Online; accessed September 5, 2023). *Between-Subjects vs. Within-Subjects Study Design*. <https://www.nngroup.com/articles/between-within-subjects/>.

- Burgees, M. (Online; accessed Augst 10, 2024). *Thousands of Corporate Secrets Were Left Exposed. This Guy Found Them All*. <https://www.wired.com/story/secret-hunting-bill-demirkapi/>.
- Busse, K., Schäfer, J., and Smith, M. (2019). Replication: No one can hack my mind revisiting a study on expert and {Non-Expert} security practices and advice. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pages 117–136.
- Cass, S. (Online; accessed November 5, 2024). *The Top Programming Languages 2024*. <https://spectrum.ieee.org/top-programming-languages-2024>.
- Chandramouli, R. (Online; accessed September 5, 2023). *Strategies for the Integration of Software Supply Chain Security in DevSecOps CI/CD pipelines*. <https://csrc.nist.gov/pubs/sp/800/204/d/ipd?ref=blog.gitguardian.com>.
- Chen, J.-C. and Huang, S.-J. (2009). An empirical analysis of the impact of software development problem factors on software maintainability. *Journal of Systems and Software*, 82(6):981–992.
- Cheng, L., Murphy-Hill, E., Canning, M., Jaspán, C., Green, C., Knight, A., Zhang, N., and Kammer, E. (2022). What improves developer productivity at google? code quality. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1302–1313.
- Christin, N., Egelman, S., Vidas, T., and Grossklags, J. (2011). It’s all about the benjamins: An empirical study on incentivizing users to ignore security advice. In *International Conference on Financial Cryptography and Data Security*, pages 16–30. Springer.
- Cidon, A., Gavish, L., Bleier, I., Korshun, N., Schweighauser, M., and Tsitkin, A. (2019). High precision detection of business email compromise. In *28th USENIX Security Symposium*, pages 1291–1307.
- Cohen, J. (1988). *Statistical power analysis for the behavioural sciences*. Hillsdale, NJ: Laurence Erlbaum Associates.
- Community, D. (Online; accessed July 1, 2024). *Build and deploy a web app with Python, Flask, and Doppler*. <https://dev.to/lordghostx/build-and-deploy-a-web-app-with-python-flask-and-doppler-9jm>.
- Cornelissen, J. P. (2020). Corporate communication: A guide to theory and practice. *Corporate Communication*, pages 1–336.
- Dahlmanns, M., Sander, C., Decker, R., Wehrle, K., Pennekamp, J., Belova, A., Bergs, T., Bodenbenner, M., Bührig-Polaczek, A., Kunze, I., et al. (2023). Secrets revealed in container images: An internet-wide study on occurrence and impact. In *ACM Transactions on Internet Technology*, number 252-266, pages 252–266. ACM.

- Dang-Pham, D., Pittayachawan, S., and Bruno, V. (2017). Why employees share information security advice? exploring the contributing factors and structural patterns of security advice sharing in the workplace. *Computers in Human Behavior*, 67:196–206.
- Denning, T., Lerner, A., Shostack, A., and Kohno, T. (2013). Control-alt-hack: the design and evaluation of a card game for computer security awareness and education. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 915–928.
- Doppler (Online; accessed June 30, 2024). *Python*. <https://docs.doppler.com/docs/vscode-python>.
- Doppler (Online; accessed September 5, 2023). *The New Era of Secrets Management*. <https://www.doppler.com/>.
- Egele, M., Stringhini, G., Kruegel, C., and Vigna, G. (2015). Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing*, 14(4):447–460.
- Elliott, V. (2018). Thinking about the coding process in qualitative data analysis. *The Qualitative Report*, 23(11):2850–2861.
- Ernst, N. A. and Robillard, M. P. (2023). A study of documentation for software architecture. *Empirical Software Engineering*, 28(5):122.
- Fagan, M. and Khan, M. M. H. (2016). 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.
- Fourné, M., Braga, D. D. A., Jancar, J., Sabt, M., Schwabe, P., Barthe, G., Fouque, P.-A., and Acar, Y. (2024). “these results must be false”: A usability evaluation of constant-time analysis tools. In *33th USENIX Security Symposium (USENIX Security 2024)*.
- Freelon, D. *ReCal2: Reliability for 2 Coders*.
- Freelon, D. *ReCal3: Reliability for 3+ Coders*.
- Fritz, C. (2011). E Morris P, J Richler J. Effect Size Estimates: Current Use, Calculations, and Interpretation. *J Exp Psychol Gen*, 8:2–18.
- Fulton, K. R., Gelles, R., McKay, A., Abdi, Y., Roberts, R., and Mazurek, M. L. (2019a). The effect of entertainment media on mental models of computer security. In *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*, pages 79–95.

- Fulton, K. R., Gelles, R., McKay, A., Abdi, Y., Roberts, R., and Mazurek, M. L. (2019b). The effect of entertainment media on mental models of computer security. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pages 79–95, Santa Clara, CA. USENIX Association.
- g2 (Online; accessed September 5, 2023). *Best Secrets Management Tools*. <https://www.g2.com/categories/secrets-management-tools>.
- Gelernter, N., Kalma, S., Magnezi, B., and Porcilan, H. (2017). The password reset MitM attack. In *2017 IEEE Symposium on Security and Privacy*, pages 251–267. IEEE.
- GitHub (Online; accessed September 5, 2023a). *Managing your account-specific secrets for GitHub Codespaces*. <https://docs.github.com/en/codespaces/managing-your-codespaces/managing-your-account-specific-secrets-for-github-codespaces>.
- GitHub (Online; accessed September 5, 2023b). *Using secrets in GitHub Actions*. <https://docs.github.com/en/actions/security-for-github-actions/security-guides/using-secrets-in-github-actions>.
- Google (Online; accessed September 5, 2023). *Google Secret Manager*. <https://cloud.google.com/secret-manager>.
- Gorski, P. L., Iacono, L. L., Wermke, D., Stransky, C., Möller, S., Acar, Y., and Fahl, S. (2018). Developers deserve security warnings, too: On the effect of integrated security advice on cryptographic {API} misuse. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, pages 265–281.
- Guri, M., Shemer, E., Shirtz, D., and Elovici, Y. (2016). Personal information leakage during password recovery of internet services. In *2016 European Intelligence and Security Informatics Conference (EISIC)*, pages 136–139. IEEE.
- Haney, J., Acar, Y., and Furman, S. (2021). "it's the company, the government, you and i": User perceptions of responsibility for smart home privacy and security. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 411–428.
- Haney, J. M., Theofanos, M., Acar, Y., and Prettyman, S. S. (2018). "we make it a big deal in the company": Security mindsets in organizations that develop cryptographic products. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, pages 357–373.
- HashiCorp (Online; accessed June 30, 2024). *hcp vault-secrets run*. <https://developer.hashicorp.com/hcp/docs/cli/commands/vault-secrets/run>.
- HashiCorp (Online; accessed September 5, 2023). *What is HCP Vault Secrets?* <https://developer.hashicorp.com/hcp/docs/vault-secrets/>.

- Head, A., Sadowski, C., Murphy-Hill, E., and Knight, A. (2018). When not to comment: Questions and tradeoffs with api documentation for c++ projects. In *Proceedings of the 40th International Conference on Software Engineering*, pages 643–653.
- Herley, C. (2009). So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 Workshop on New Security Paradigms Workshop*, pages 133–144.
- Herley, C. (2013). More is not the answer. *IEEE Security & Privacy*, 12(1):14–19.
- Huh, J. H., Kim, H., Rayala, S. S., Bobba, R. B., and Beznosov, K. (2017). I’m too busy to reset my linkedin password: On the effectiveness of password reset emails. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 387–391.
- Hunt, T. (Online; accessed March, 2020). *Pwned Websites*. <http://haveibeenpwned.com/PwnedWebsites/>.
- IBM (Online; accessed March, 2020). *IBM SPSS software*. <https://www.ibm.com/analytics/spss-statistics-software>.
- Indela, S., Kulkarni, M., Nayak, K., and Dumitras, T. (2016). Helping johnny encrypt: Toward semantic interfaces for cryptographic frameworks. In *Proceedings of the 2016 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, pages 180–196.
- Infisical (Online; accessed September 5, 2023). *Open Source Secret Management*. <https://infisical.com/>.
- itnews (Online; accessed September 5, 2023). *AWS urges developers to scrub GitHub of secret keys*. <https://www.itnews.com.au/news/aws-urges-developers-to-scrub-github-of-secret-keys-375785>.
- Jackson, M. (Online; accessed September 5, 2023). *8.5% of Docker Images Expose API and Private Keys*. <https://blog.gitguardian.com/8docker-images-api-and-private-keys/>.
- Jansen, P. (Online; accessed November 5, 2024). *TIOBE Index for November 2024*. <https://www.tiobe.com/tiobe-index/>.
- Kanniah, S. L. and Mahrin, M. N. (2016). A review on factors influencing implementation of secure software development practices. *International Journal of Computer and Systems Engineering*, 10(8):3032–3039.
- Karimi, H., VanDam, C., Ye, L., and Tang, J. (2018). End-to-end compromised account detection. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 314–321. IEEE.

- Katz, E. (Online; accessed September 5, 2023). *5 Ways to Prevent Secrets Sprawl*. <https://spectralops.io/blog/5-ways-to-prevent-secrets-sprawl/>.
- Krause, A., Klemmer, J. H., Huaman, N., Wermke, D., Acar, Y., and Fahl, S. (2023). Pushed by accident: A {Mixed-Methods} study on strategies of handling secret information in source code repositories. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2527–2544.
- Krüger, S., Nadi, S., Reif, M., Ali, K., Mezini, M., Bodden, E., Göpfert, F., Günther, F., Weinert, C., Demmler, D., et al. (2017). Cognicrypt: Supporting developers in using cryptography. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 931–936. IEEE.
- Krüger, S., Reif, M., Wickert, A.-K., Nadi, S., Ali, K., Bodden, E., Acar, Y., Mezini, M., and Fahl, S. (2023). Securing your crypto-api usage through tool support—a usability study. In *2023 IEEE Secure Development Conference (SecDev)*, pages 14–25. IEEE.
- Kumaraguru, P. (2009). *Phishguru: a system for educating users about semantic attacks*. Carnegie Mellon University.
- Kumaraguru, P., Cranshaw, J., Acquisti, A., Cranor, L., Hong, J., Blair, M. A., and Pham, T. (2009). School of phish: a real-world evaluation of anti-phishing training. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12.
- Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and anovas. *Frontiers in psychology*, 4:863.
- Lazar, J., Feng, J. H., and Hochheiser, H. (2017). *Research methods in human-computer interaction*. Morgan Kaufmann.
- Leonhardt, M. (2019.). *The 5 biggest data hacks of 2019*. <https://www.cnbc.com/2019/12/17/the-5-biggest-data-hacks-of-2019.html>.
- Lethbridge, T. C., Singer, J., and Forward, A. (2003). How software engineers use documentation: The state of the practice. *IEEE software*, 20(6):35–39.
- Matsiako, V. (Online; accessed September 5, 2023). *Top-10 Secret Management Tools in 2024*. <https://infisical.com/blog/best-secret-management-tools>.
- Mayer, P., Zou, Y., Schaub, F., and Aviv, A. J. (2021). " Now I'm a bit angry:" Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them. In *30th USENIX Security Symposium*.
- Mcdaniel, D. (Online; accessed September 5, 2023a). *A look at the future of supply chain and national security: Updates from CISA and NIST*. <https://blog.gitguardian.com/software-supply-chain-security-updates-from-cisa-and-nist/>.

- Mcdaniel, D. (Online; accessed September 5, 2023b). *Toyota Suffered a Data Breach by Accidentally Exposing A Secret Key Publicly On GitHub*. <https://blog.gitguardian.com/toyota-accidently-exposed-a-secret-key-publicly-on-github-for-five-years/>.
- McDonald, N., Schoenebeck, S., and Forte, A. (2019a). Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. In *ACM on Human-Computer Interaction*, page 72.
- McDonald, N., Schoenebeck, S., and Forte, A. (2019b). Reliability and inter-rater reliability in qualitative research: Norms and guidelines for cscw and hci practice. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–23.
- Medium (Online; accessed July 1, 2024). *Injecting secrets to Kubernetes containers from the Doppler secrets manager*. <https://medium.com/@peterkracik/injecting-secrets-to-kubernetes-containers-from-the-doppler-secrets-manager-ef491a20f45b>.
- Mekhail, C., Zhang-Kennedy, L., and Chiasson, S. (2014). Visualizations to teach about mobile online privacy. In *Persuasive Technology Conference (poster)*.
- Meli, M., McNiece, M. R., and Reaves, B. (2019). How bad can it get? characterizing secret leakage in public github repositories. In *NDSS*.
- Microsoft (Online; accessed June 30, 2024). *Visual Studio Code*. <https://code.visualstudio.com/>.
- Microsoft (Online; accessed September 5, 2023). *Azure Key Vault*. <https://learn.microsoft.com/en-us/azure/key-vault/>.
- Middleton, J., Murphy-Hill, E., and Stolee, K. T. (2020). Data analysts and their software practices: A profile of the sabermetrics community and beyond. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–27.
- Murphy-Hill, E., Jaspán, C., Sadowski, C., Shepherd, D., Phillips, M., Winter, C., Knight, A., Smith, E., and Jorde, M. (2019). What predicts software developers’ productivity? *IEEE Transactions on Software Engineering*, 47(3):582–594.
- Nassif, M., Hernandez, A., Sridharan, A., and Robillard, M. P. (2021). Generating unit tests for documentation. *IEEE Transactions on Software Engineering*, 48(9):3268–3279.
- Nassif, M., Horlacher, Z., and Robillard, M. P. (2022). Casdoc: unobtrusive explanations in code examples. In *Proceedings of the 30th IEEE/ACM international conference on program comprehension*, pages 631–635.

- Nassif, M. and Robillard, M. P. (2023a). A field study of developer documentation format. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–7.
- Nassif, M. and Robillard, M. P. (2023b). Identifying concepts in software projects. *IEEE Transactions on Software Engineering*, 49(7):3660–3674.
- Nassif, M. and Robillard, M. P. (2023c). Non linear software documentation with interactive code examples. *arXiv preprint arXiv:2311.18057*.
- Neil, L., Acar, Y., and Reaves, B. (2020). Investigating Web Service Account Remediation Advice. In *Who Are You?! Adventures in Authentication Workshop, WAY '20*, pages 1–6, Virtual Conference.
- Neil, L., Bouma-Sims, E., Lafontaine, E., Acar, Y., and Reaves, B. (2021). Investigating web service account remediation advice. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 359–376. USENIX Association.
- News, U. (Online; accessed February, 22, 2022). Best national university rankings. https://www.usnews.com/best-colleges/rankings/national-universities?_mode=table. [Online; accessed February 23, 2022].
- Nicholson, J., Coventry, L., and Briggs, P. (2019). "if it's important it will be a headline" cybersecurity information seeking in older adults. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11.
- Noda, A., Storey, M.-A., Forsgren, N., and Greiler, M. (2023). Devex: What actually drives productivity: The developer-centric approach to measuring and improving productivity. *Queue*, 21(2):35–53.
- OBS (Online; accessed February, 22, 2022). Open broadcaster software. <https://obsproject.com/wiki/OBS-Studio-Overview>. [Online; accessed February 23, 2022].
- of State, U. D. (Online; accessed February 23, 2022). *Fraud Warning*. <https://travel.state.gov/content/travel/en/us-visas.html/>.
- Oh, S., Lee, K., Park, S., Kim, D., and Kim, H. (2024). Poisoned chatgpt finds work for idle hands: Exploring developers' coding practices with insecure suggestions from poisoned ai models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 1141–1159. IEEE.
- Onaolapo, J., Mariconti, E., and Stringhini, G. (2016). What happens after you are pwnd: Understanding the use of leaked webmail credentials in the wild. In *Proceedings of the 2016 Internet Measurement Conference*, pages 65–79.

- Pal, B., Daniel, T., Chatterjee, R., and Ristenpart, T. (2019). Beyond credential stuffing: Password similarity models using neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 417–434. IEEE.
- Parkin, S., Driss, S., Krol, K., and Sasse, M. A. (2015). Assessing the user experience of password reset policies in a university. In *International Conference on Passwords*, pages 21–38. Springer.
- Parnin, C., Treude, C., Grammel, L., and Storey, M.-A. (2012). Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. *Georgia Institute of Technology, Tech. Rep*, 11.
- Peng, P., Xu, C., Quinn, L., Hu, H., Viswanath, B., and Wang, G. (2019). What happens after you leak your password: Understanding credential sharing on phishing sites. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 181–192.
- Pfeffer, K., Mai, A., Weippl, E., Rader, E., and Krombholz, K. (2022). Replication: Stories as informal lessons about security. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, pages 1–18.
- Pochat, V. L., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., and Joosen, W. (2018). Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156*.
- PyPI (Online; accessed June 30, 2024). *doppler-env 0.3.1*. <https://pypi.org/project/doppler-env/>.
- Qualtrics (Online; accessed February, 22, 2022). qualtrics. <https://www.qualtrics.com/>. [Online; accessed February 23, 2022].
- Rader, E. and Wash, R. (2015). Identifying patterns in informal sources of security information. *Journal of Cybersecurity*, 1(1):121–144.
- Rader, E., Wash, R., and Brooks, B. (2012). Stories as informal lessons about security. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, pages 1–17.
- Redmiles, E. M., Kross, S., and Mazurek, M. L. (2016a). How i learned to be secure: a census-representative survey of security advice sources and behavior. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 666–677.
- Redmiles, E. M., Kross, S., and Mazurek, M. L. (2017). Where is the digital divide? a survey of security, privacy, and socioeconomics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 931–936.

- Redmiles, E. M., Malone, A. R., and Mazurek, M. L. (2016b). I think they're trying to tell me something: Advice sources and selection for digital security. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 272–288. IEEE.
- Redmiles, E. M., Mazurek, M. L., and Dickerson, J. P. (2018). Dancing pigs or externalities? measuring the rationality of security decisions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 215–232.
- Redmiles, E. M., Warford, N., Jayanti, A., Koneru, A., Kross, S., Morales, M., Stevens, R., and Mazurek, M. L. (2020). A comprehensive quality evaluation of security and privacy advice on the web. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 89–108.
- Reeder, R. W., Ion, I., and Consolvo, S. (2017). 152 simple steps to stay safe online: Security advice for non-tech-savvy users. *IEEE Security & Privacy*, 15(5):55–64.
- ReversingLabs (Online; accessed September 5, 2023). *Secrets Exposed: How to mitigate risk from secrets leaks — and prevent future breaches*. <https://www.reversinglabs.com/blog/secure-your-development-secrets-3-essential-steps>.
- Robillard, M. P. (2009). What makes apis hard to learn? answers from developers. *IEEE software*, 26(6):27–34.
- Robillard, M. P. and Chhetri, Y. B. (2015). Recommending reference api documentation. *Empirical Software Engineering*, 20(6):1558–1586.
- Robinson, D., Ernst, N. A., Vargas, E. L., and Storey, M.-A. D. (2022). Error identification strategies for python jupyter notebooks. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, pages 253–263.
- Ruan, X., Wu, Z., Wang, H., and Jajodia, S. (2015). Profiling online social behaviors for compromised account detection. *IEEE transactions on information forensics and security*, 11(1):176–187.
- Saunders, B., Sim, J., Kingstone, T., Baker, S., Waterfield, J., Bartlam, B., Burroughs, H., and Jinks, C. (2018). Saturation in qualitative research: exploring its conceptualization and operationalization. *Quality & quantity*, 52(4):1893–1907.
- Schechter, S., Brush, A. B., and Egelman, S. (2009). It's no secret. measuring the security and reliability of authentication via “secret” questions. In *30th IEEE Symposium on Security and Privacy*, pages 375–390. IEEE.
- Segal, J. (2007). Some problems of professional end user developers. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, pages 111–118. IEEE.

- Shay, R., Ion, I., Reeder, R. W., and Consolvo, S. (2014). " My religious aunt asked why I was trying to sell her viagra" experiences with account hijacking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2657–2666.
- Sheng, S., Magnien, B., Kumaraguru, P., Acquisti, A., Cranor, L. F., Hong, J., and Nunge, E. (2007). 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*, pages 88–99.
- Sinha, V. S., Saha, D., Dhoolia, P., Padhye, R., and Mani, S. (2015). Detecting and mitigating secret-key leaks in source code repositories. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 396–400. IEEE.
- Smith, J., Do, L. N. Q., and Murphy-Hill, E. (2020). Why can't johnny fix vulnerabilities: A usability evaluation of static analysis tools for security. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 221–238.
- Sohan, S., Maurer, F., Anslow, C., and Robillard, M. P. (2017). A study of the effectiveness of usage examples in rest api documentation. In *2017 IEEE symposium on visual languages and human-centric computing (VL/HCC)*, pages 53–61. IEEE.
- Stackoverflow (Online; accessed November 5, 2024). *2024 Developer Survey*. <https://survey.stackoverflow.co/2024/technology#2-programming-scripting-and-markup-languages>.
- Storey, M.-A., Russo, D., Novielli, N., Kobayashi, T., and Wang, D. (2024). A disruptive research playbook for studying disruptive innovations. *ACM Transactions on Software Engineering and Methodology*.
- Storey, M.-A., Zimmermann, T., Bird, C., Czerwonka, J., Murphy, B., and Kalliamvakou, E. (2019). Towards a theory of software developer job satisfaction and perceived productivity. *IEEE Transactions on Software Engineering*, 47(10):2125–2142.
- Subramanian, S., Inozemtseva, L., and Holmes, R. (2014). Live api documentation. In *Proceedings of the 36th international conference on software engineering*, pages 643–652.
- Tahaei, M. and Vaniea, K. (2019). A survey on developer-centred security. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 129–138. IEEE.
- Thomas, K., Li, F., Zand, A., Barrett, J., Ranieri, J., Invernizzi, L., Markov, Y., Comanescu, O., Eranti, V., Moscicki, A., et al. (2017). Data breaches, phishing, or malware? Understanding the risks of stolen credentials. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1421–1434.

- Thomas, K., Pullman, J., Yeo, K., Raghunathan, A., Kelley, P. G., Invernizzi, L., Benko, B., Pietraszek, T., Patel, S., Boneh, D., et al. (2019). Protecting accounts from credential stuffing with password breach alerting. In *Proceedings of the 28th USENIX Security Symposium*, pages 1556–1571.
- Treude, C. and Robillard, M. P. (2016). Augmenting api documentation with insights from stack overflow. In *Proceedings of the 38th International Conference on Software Engineering*, pages 392–403.
- Turner, S., Nurse, J., and Li, S. (2021). When googling it doesn't work: The challenge of finding security advice for smart home devices. In *International Symposium on Human Aspects of Information Security and Assurance*, pages 115–126. Springer.
- Uddin, G. and Robillard, M. P. (2015). How api documentation fails. *Ieee software*, 32(4):68–75.
- Upwork (Online; accessed February 23, 2022). Upwork. <https://www.upwork.com/>.
- VanDam, C., Tang, J., and Tan, P.-N. (2017). Understanding compromised accounts on twitter. In *Proceedings of the International Conference on Web Intelligence*, pages 737–744.
- Wagner, S. and Murphy-Hill, E. (2019). Factors that influence productivity: A checklist. *Rethinking productivity in software engineering*, pages 69–84.
- Wang, K. C. and Reiter, M. K. (2020). Detecting stuffing of a user's credentials at her own accounts. In *29th USENIX Security Symposium*, pages 2201–2218.
- Wen, F., Nagy, C., Bavota, G., and Lanza, M. (2019). A large-scale empirical study on code-comment inconsistencies. In *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*, pages 53–64. IEEE.
- Williams, L., Meneely, A., and Shipley, G. (2010). Protection poker: The new software security" game". *IEEE Security & Privacy*, 8(3):14–20.
- Witschey, J., Zielinska, O., Welk, A., Murphy-Hill, E., Mayhorn, C., and Zimmermann, T. (2015). Quantifying developers' adoption of security tools. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 260–271.
- Xie, J., Lipford, H. R., and Chu, B. (2011). Why do programmers make security errors? In *2011 IEEE symposium on visual languages and human-centric computing (VL/HCC)*, pages 161–164. IEEE.
- Zhang-Kennedy, L., Chiasson, S., and Biddle, R. (2016). The role of instructional design in persuasion: A comics approach for improving cybersecurity. *International Journal of Human-Computer Interaction*, 32(3):215–257.

APPENDICES

APPENDIX

A

INVESTIGATION OF ACCOUNT
REMEDICATION FULL CODEBOOK AND
WEB SERVICES STUDIED

A.1 Codebook

Table A.1: Codebook for Compromise Discovery codes among web services.

Compromise Discovery		
Codes	Code Explanations	Examples
Billing/finance issues	Unwanted changes in financial or billing settings/standings or unauthorized credit card charges.	You see charges or notices for purchases that you didn't make.
Email changed	Observe any email associated with account has been changed.	What do I do if someone changed my email address?
Explicit notification	Service notifies you of login or possible compromise by email or other factor. Check this if the service sends emails about new logins.	You receive an email or notification that your Apple ID was used to sign in to a device you don't recognize or did not sign in to recently (for example, "Your Apple ID was used to sign in to iCloud on a Windows PC").
Account locked by provider	Cannot access account due to account being locked or disabled.	For your protection, we may place a temporary hold on your account.
Account otherwise unavailable	Account is not accessible due to circumstances outside of provider locking account.	You can't sign in for another reason.
Observed unauthorized logins	Includes if "observation" is due to a notification from the service, but not exclusively.	You see logins from unexpected locations on your recent activity page.
Password changed	Observe password associated with account has been changed.	Someone changed the password on my Etsy account.
Social media or third party account connected	Unwanted social media becomes associated with account.	A malicious application has been given access to your account.
Unauthorized/suspicious activity	Including changed content on streaming sites, but must be more than login. For example messages, friend requests, playlists, etc.	If you notice unfamiliar activity on your Google Account, someone else might be using it without your permission. Use the info below to help spot suspicious activity.

Table A.2: Codebook for Account Recovery codes among web services.

Account Recovery		
Codes	Code Explanations	Examples
Customer service process	Engage with service customer support (chat client, form, email, etc) to regain access/reset password.	If you can't access your account and believe that someone else has accessed it, complete the form and after receiving it we'll verify that it's your account and then help you regain access.
Password reset	Initiate a password reset challenge or go through password change process.	Change your password immediately.
Run endpoint security	Run external security applications on computer to stop a suspected <i>ongoing</i> attack.	If you see any successful sign-in that you do not recognize, run a scan with your security software and remove any malware you find.

Table A.3: Codebook for Limiting Access codes among web services.

Limiting Access		
Codes	Code Explanations	Examples
Remove third party access	Disallow external third party applications (including social media) from accessing account.	Revoke access to any suspicious third-party apps.
Review active session	Review activity/logs for currently active sessions to see if compromise is ongoing.	Review your active sessions to see all the places you're signed into LinkedIn right now.
Sign out everywhere (specific function)	Logs out <i>all</i> instances of account (not just one or a few).	We recommend to log out of all computers from your phone.
Sign out of unknown session	Logs out of individual unrecognized instances of account.	If your account does get hacked, you can remove any trusted devices that you didn't log in to yourself.

Table A.4: Codebook for Service Restoration codes among web services.

Service Restoration		
Codes	Code Explanations	Examples
Customer service process	Engage with service customer support (chat client, form, email, etc.) to help restore data etc.	Contact us for help removing unauthorized bids or listings.
Fix logs of past viewing/activity/content history	For example, viewing history, input to recommendations, past purchases.	Review Order history for unrecognized charges.
Review and/or remove activities/content	For example, deleting friends you didn't add, messages you didn't write.	Delete any resources on your account that you didn't create, such as EC2 instances and AMIs, EBS volumes and snapshots, and IAM users.
Verify settings	User should verify security, privacy, or account settings.	Review your general account settings to make sure all other information is correct.
Verify user information	User should check the identifying information for users (email, name, address, or payment info like credit card number).	Verify that the email address and mobile number associated with your account are accurate in Snapchat settings.

Table A.5: Codebook for Prevention codes among web services.

Prevention		
Codes	Code Explanations	Examples
Advice about secure email	Describes advice on suspicious emails, phishing, etc.	Phishing is when someone tries to trick you into giving up your Twitter username, email address or phone number and password, usually so they can send out spam from your account.
Always log out on shared devices	Always log out shared instances of account.	Sign out of public computers- -Always sign out of your accounts when you're done.
Check/modify related accounts	For example, email accounts, shared passwords, etc.	Check your personal email account(s) tied to your account to ensure their security.
Enable 2FA	Enable any 2FA for every login attempt.	Enable Two-Factor Authentication (2FA).
Enable phone-based recovery	Enable ability to <i>recover</i> account/credentials by using a phone number as a second factor.	Add a recovery phone number to your account so that you can get back into your account faster and keep your account more secure.
Keep software up to date	Catchall: any application/program/devices/software up to date with current updates.	Regularly patch, update, and secure the operating system and applications on your instance.
Password advice: strong, unique, change frequently	Catchall for any password advice (good bad or otherwise).	Create a strong password. Make it unique: Do not reuse an existing password when setting up an account for PlayStation Network.
Physical security	Catchall for any advice to maintain physical security of devices, environment, etc.	Don't leave your devices unlocked or unattended where anyone can use it.
Remove access to third party apps	Prompted to disallow external third party applications from accessing account.	Remove suspicious applications or browser add-ons.
Run endpoint security solutions	Run external security programs/applications on computer to prevent <i>future</i> attacks.	Always use an antivirus program to check the files you receive from other people.
Sign out of devices	Log out of <i>individual</i> devices that have instances of account.	Log out when you are done.

A.2 Web Services Studied

Very Popular Websites		Less Popular Websites	
Ranking	Website	Ranking	Website
1	google.com	524	hootsuite.com
2	facebook.com	542	ox.ac.uk
3	youtube.com	547	umn.edu
4	microsoft.com	559	uci.edu
5	twitter.com	568	ucla.edu
7	instagram.com	575	att.com
9	netflix.com	578	snapchat.com
10	linkedin.com	608	uchicago.edu
13	wikipedia.org	620	playstation.com
14	apple.com	635	xfinity.com
18	yahoo.com	658	parallels.com
23	pinterest.com	669	epicgames.com
25	vimeo.com	682	fidelity.com
28	reddit.com	730	ning.com
40	amazonaws.com	776	verizon.com
44	tumblr.com	785	uber.com
45	godaddy.com	795	msu.edu
51	skype.com	806	ea.com
55	whatsapp.com	836	northwestern.edu
56	dropbox.com	837	crunchyroll.com
58	soundcloud.com	886	arizona.edu
61	myshopify.com	904	wattpad.com
67	twitch.tv	917	stripe.com
79	spotify.com	932	namecheap.com
81	paypal.com	942	xbox.com
93	cloudflare.com		
94	ebay.com		
117	etsy.com		
170	aol.com		

183	fandom.com
188	walmart.com
209	yelp.com

Table A.6: List of web services studies for account remediation advice.

A.3 Data

Our annotated advice is available through this [link](#).

APPENDIX

B

INTERVIEWS WITH SECURITY ADVICE AUTHORS INTERVIEW GUIDE AND HIGH LEVEL CODES

B.1 Interview Guide

1. Ice Breaker Questions:
 - (a) When and where did you learn to write general security advice?
 - (b) What is your current occupation?
 - (c) Can you describe the type of company you worked for when you wrote general security advice?
2. Can you tell me about how security advice gets made and distributed at your organization?

- (a) Is there a decision making model for the creation of security advice?
 - (b) Are there any external sources used to provide sample advice that gets posted?
3. Can you tell me about the people or roles involved in the process?
- (a) Does a chain of command or hierarchy exist within the parties?
 - (b) Are all of these parties involved with the company or external?
 - (c) What is typically the experience or knowledge of parties in regards to computer security?"
4. Are there particular areas that are prioritized or discussed more in depth within the general security advice?
- (a) If so, what is the reason for this prioritization or focus into this area?
 - (b) Are there any areas that are intentionally excluded from being covered in the security advice?
 - (c) If so, what is the reason for not writing advice for this specific area?
5. Is the general security advice regularly updated or reviewed?
- (a) If so, what systems or procedures are in place to update/review the advice?
 - (b) Were these systems/procedures always in place, or did an event or policy create them?
 - (c) If possible to comment, are there legal practices or regulations that prompt the creation and/or regulation of the advice?
6. Is your company's legal department involved in the creation or even discussion of the general security advice?
- (a) If so and you are able to comment, are they able to edit or create any parts of the advice, or even recommend certain areas be covered?
7. If possible, can you comment on how much responsibility your organization claims in assisting in general security?
- (a) How much of the advice is well-meant, or meant to limit the reliability/responsibility of the service in security matters with general security?

8. Does your company have a team or group of individuals that handle general security internally?
 - (a) Are they external workers?
 - (b) Do they have expert experience in computer security?
9. When creating the general security advice, is there a thought process as to how actionable or practical the advice may be for the typical user?
10. These last questions are more so geared to your own experiences when creating the advice.
 - (a) Are there any tasks completed during general security advice creation/revisions that are challenging or time consuming?
 - (b) Have you ever thought about how general security advice for your company, or overall can be improved?

B.2 Codebook

High Level Codes	
Codes	Code Explanations
1a. Learn to write Advice	How the participant first learned to write general security advice.
1b. Occupational Role	Occupations for participants during the time they wrote general security advice.
1c. Companies	Places where the participant worked for and wrote the advice.
2a. Formal Writing Process	Any formal or structured process (Gap Analysis, SLA, defining scope, etc) used for writing advice.
2b. Informal Writing Process	Advice writing that is not dependent on any formal process. Rather, it is written in an informal or non-structured writing process.
2c. Legal or Non Legal Guidelines	Mandates, regulations, laws, or frameworks that were used to influence the advice. These are not solely or specifically technical, but apply to a wider range of compliance standards.
2d. Technical, Security Standards	Advice content is influenced by technical and/or security standards.
2e. External Entities	External entities (organization, group, company, etc) that authors seek for guidance on advice writing.
3a. Background, Experience	Backgrounds of fellow workers/teammates of advice authors.
3b. External Company Party Collaboration	Parties outside the primary advice construction group that collaborate in the advice writing process (outside or external to the company).
3c. Internal Company Party Collaboration	Parties outside the primary advice construction group that collaborate in the advice writing process (within the company).
3d. Writers	The number of people specified by the participant who helps physically write the advice.

4a. Most Prioritized Advice	Most common/prioritized topics of advice written.
4b. Least Prioritized Advice	Least common/prioritized topics of general security advice.
4c. Reasons Advice is Prioritized	Reasons or events that would cause the creation of general security advice.
4d. Reasons Advice is not Prioritized	Reasons certain advice has not been covered as much or prioritized.
5a. Revision Process	Processes and reasons to revise advice.
6. Company's legal department	Company's legal department involvement within the advice writing process.
7. Responsibilities	Responsibilities claimed by participant companies when creating the advice.
8. Internal Support	Support for clients that is internal or technical (not advice).
9. Advice Usability Thought Process	Though process or methods of improving actionability/usability of the advice.
10a. Challenges	Challenges with writing the advice.
10b. Improvements	Authors' opinions of how the advice writing process could be improved.

Table B.1: Full list of high level codes from codebook for security advice interviews.

APPENDIX

C

MIXED-METHODS STUDY WITH NEW SMT USERS INTERVIEW QUESTIONS AND CODEBOOK

C.1 Interview Guide

Warm-up Questions: Prior experience with managing secrets

1. Do you have any prior experience with managing secrets? (E.g., managing secrets for an industry role? Academic project?)
2. What type of secrets did you manage?
3. Did you use any specific tools or perform specific management practices?

Task 1: Overall Experience

1. How would you describe your overall experience of performing this task?
2. Is there anything that you did not like or find difficult for this task?
 - a) Vice Versa?

Task 1: Negative Emotions

1. Were there any parts during this task exercise where you experienced any negative emotions? *(This can include annoyance, confusion, fatigue, frustration, or any form of disapproval/dissatisfaction with either the tool, the advice you looked at, or the task itself.)*
 - a) If so, please explain why you felt those emotions.

Task 1: Tool Documentation

1. Did you experience any difficulties or challenges when using code snippets, examples, or explanatory text from the tool documentation as resources to complete the task?
 - a) If so, why? What made using these specific qualities of the content challenging?
2. Did you experience difficulty in mentally connecting the process for what you were assigned to do in the task with how the respective content is provided in the tool documentation? *(To rephrase: Was it challenging to understand how to complete the task assigned to you as you were reading the content provided in the tool documentation?)*
 - a) If so, please explain
3. Did you experience any challenges in finding the information that allowed you to complete or progress through the task?
 - a) If so, please explain what challenges you faced in finding the information.
4. Did the structure of the tool documentation content (e.g., section headers, usage of formatting like bullet points or lists, toolbar format, etc) present any challenges for you in finding information and completing the task?
 - a) If so, please explain. and b) Vice Versa?
5. Did the format of the advice content (as it relates to text, pictures, videos) present any challenges to you or make it harder to complete the task?
 - a) If so, please explain.
 - b) Vice Versa?

Task 1: Secondary Sources

1. If you looked up advice from sources external to the tool documentation, why did you search for other resources?
 - a) What specifically did you look for in secondary sources that were not present and/or made clear within the help documentation?
2. Were there any external sources that you preferred or found more helpful towards completing the task than the tool documentation?
 - a) If so, what were these sources and why did you find them more helpful?

Task 2: Overall Experience

1. How would you describe your overall experience of performing this task?

2. Is there anything that you did not like or find difficult for this task?
 - a) Vice Versa?

Task 2: Negative Emotions

1. Were there any parts during this task exercise where you experienced any negative emotions? (*This can include annoyance, confusion, fatigue, frustration, or any form of disapproval/dissatisfaction with either the tool, the advice you looked at, or the task itself.*)
 - a) If so, please explain why you felt those emotions.

Task 2: Tool Documentation

1. Did you experience any difficulties or challenges when using code snippets, examples, or explanatory text from the tool documentation as resources to complete the task?
 - a) If so, why? What made using these specific qualities of the content challenging?
2. Did you experience difficulty in mentally connecting the process for what you were assigned to do in the task with how the respective content is provided in the tool documentation? (*To rephrase: Was it challenging to understand how to complete the task assigned to you as you were reading the content provided in the tool documentation?*)
 - a) If so, please explain
3. Did you experience any challenges in finding the information that allowed you to complete or progress through the task?
 - a) If so, please explain what challenges you faced in finding the information.
4. Did the structure of the tool documentation content (e.g., section headers, usage of formatting like bullet points or lists, toolbar format, etc) present any challenges for you in finding information and completing the task?
 - a) If so, please explain. and b) Vice Versa?
5. Did the format of the advice content (as it relates to text, pictures, videos) present any challenges to you or make it harder to complete the task?
 - a) If so, please explain. and b) Vice Versa?

Task 2: Secondary Sources

1. If you looked up advice from sources external to the tool documentation, why did you search for other resources?
 - a) What specifically did you look for in secondary sources that were not present and/or made clear within the help documentation?
2. Were there any external sources that you preferred or found more helpful towards completing the task than the tool documentation?
 - a) If so, what were these sources and why did you find them more helpful?

Wrap-up Questions

1. Could you see yourself using this SMT for managing secrets within your own workflow? Or for your own projects?
 - a) If not, can you please explain why not? and b) If yes, please explain, specifically, what features or aspects help with the adoption.
2. Do you have any preferences for the types of sources you consult when needing assistance for technical tasks like the ones today? (including online or offline)
3. Do you have any preferences for how you typically like to view advice from help resources? (e.g., just text, media included, etc)
4. Do you have any last comments or questions about SMTs, tool documentation, or the study in general?

C.2 Interview Codebook

Table C.1: Interview Codebook of Participants' Experiences and Wrap-up.

Codes	Definitions
Prior Orgs. and Roles	Types of orgs(e.g.,industry, academic, govt) and roles (e.g., full-time, intern) related to managing secrets (e.g., tokens, backend APIs)
Secrets Managed	Types of secrets managed
Tools and Practices	Specific tools or practices used
Tool Usability Feedback	Reasons about using the tool in future and why/why not?
General Source Lookup	Preferred sources for general technical problems
Advice Format Preference	Preferred format for online advice (e.g., just text, media included, etc)
Last Comments	Any last comments or questions about the study
Security Implications	Concerns, comments, or observations about security of tool or task(security implications)
Recommendations	Voicings of recommendation or suggestion about tool or doc helped to finish task
General Tool Comments	Feedback on tool features (e.g., save button, UI design, copy/paste functionality)
Chat GPT/AI Comments	Responses mentioning usage of ChatGPT
Google Docs Advice Comments	Comments on using provided Google docs links
Web Portal or Dashboard Advice	Comments on web portal/dashboard (tasks or sections)

Table C.2: Interview Codebook of Tasks 1 and 2.

Codes	Definitions
Overall Experience	Overall experience to complete the task
Subcode: Overall	Captures general responses without specific positive or negative details
Subcode: Negatives	Specific difficulties, challenges, or dislikes related to the task
Subcode: Positives	Aspects participant liked, found easy about the task, documentation
Negative Emotions	Expressions of negative emotions (e.g., confusion, annoyance, frustration, fatigue, dissatisfaction) noted directly or during responses
Documentation Content Challenges	Issues with content such as code snippets, commands, UI examples, explanatory text, images, or videos
Subcode: Challenges	Issues or dislikes participants encountered when using content from the main help documentation to complete the task
Subcode: Helpfulness	Reasons the content was helpful or aspects the participant liked
Task and Documentation Mental Connection	Difficulty of linking the task description and goal with the help documentation
Subcode: Yes	If a participant says yes, and the reasoning why
Subcode: No	If a participant says no, and the reasoning why
Structure and Formatting Challenges	Issues or dislikes related to the structure or formatting of the main help documentation
Subcode: Finding Helpful Advice	Challenges of participant broadly for finding advice
Subcode: Structure	Challenges with the structure, such as page format, section headers, bullet points, tabs, or toolbars, encountered while completing the task
Subcode: Format	Challenges with the format, including text, images, videos, or other media, encountered while completing the task
Subcode: Helpfulness	Positive aspects of the structure, format, or advice that helped the participant complete the task
Secondary Source Lookup	Mentions of secondary sources outside the materials provided in the study
Subcode: Lookup	Reasons for accessing secondary sources and what the participant was seeking
Subcode: Not Helpful	Reasons why the participant found secondary sources unhelpful or challenges encountered
Subcode: Helpfulness	Responses for if the participant found the secondary sources more helpful and why/why not?