The Research-Practice Gap in User Authentication

Kevin Lee

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE BY
THE DEPARTMENT OF
COMPUTER SCIENCE

Adviser: Arvind Narayanan

September 2022

Abstract

The gap between user authentication research and practice has led to weaknesses in critical, widely-deployed systems used by millions of people. In these systems, policy and process vulnerabilities—not software vulnerabilities—allow UI-bound, low-tech adversaries to exploit weaknesses to threaten user safety. The disconnect is caused partly by practice failing to heed advice from research. But it is also caused by research not understanding the practical constraints of these systems, while discouraging studies that try to do so. Ultimately, users are the ones who suffer when these weaknesses remain undiscovered.

Here, we studied user authentication practices that were not necessarily cutting-edge, but broadly impacted user safety. We identified security policy and process flaws, quantified the risk of harm to users through manual measurements, and called for policy solutions to mitigate the risks. More broadly, we honed a methodology through these studies which can potentially bridge the research-practice gap in user authentication as well as in other topics in information security.

First, we studied call center authentication for SIM swap requests at mobile carriers. We found flaws in their authentication policy and processes which could facilitate SIM swap attacks. Furthermore, we found that most websites did not stand up well against SIM swaps, demonstrating that users' accounts could easily be hijacked. Our results have influenced policy changes at carriers and websites, and have motivated ongoing rulemaking by the FCC.

Next, we studied security and privacy risks of phone number recycling in the U.S. at mobile carriers. We found that most numbers we sampled were recycled and vulnerable to attacks on previous owners, while carriers had design weaknesses that could facilitate attacks. We have raised awareness about the risks of number recycling at carriers, and have communicated a practical constraint of SMS-based authentication to the research community.

Finally, we studied password policies of top websites. Despite well-established recommendations from research, we found few websites actually following them, which could put accounts at risk of password compromise. We hypothesized reasons why these websites were not following best practices, and discussed ways the research community could engage website system administrators to bridge the research-practice gap.

Acknowledgments

With sincerity and humility I thank:

- My advisor, Arvind Narayanan, for his unwavering guidance in these past four years. He has taught me to become a better researcher where it counts; most importantly, in pursuing research that carries societal benefit. Thank you for your support, and for believing in me when I oftentimes did not believe in myself.

- Jonathan Mayer, for being a second advisor throughout my Ph.D. His persistent support in these studies is the reason my research has been able to make an impact on society. I am grateful to have found my research direction while taking his seminar, *Computer Science for Public Policy and Law*.

- Prateek Mittal, Andrés Monroy-Hernández, and Thomas Ristenpart, for completing my dissertation committee. Each of you have given me unique perspectives that have made computer science research so much more enjoyable. Thank you all for the positive influence you have had on my dissertation.

- Mihir Kshirsagar for his guidance on my research outreach. I am especially thankful for the dedication he put into preparing my SIM swaps research for various stakeholders.

- Malte Möser, for being a longtime mentor, devoted collaborator, and close friend. I am forever grateful to him for his enthusiasm in meeting with me and showing me the ropes while I was an undergraduate student, all while he was adjusting to his first semester as a Ph.D. student. His guidance ultimately paved the way for me to pursue a Ph.D. at Princeton myself, and his friendship made the stress of graduate school so much more bearable, especially during the COVID-19 pandemic.

- My colleague, Ben Kaiser, who has been there from the start of my Ph.D. Ben has devoted countless hours to collaborating with me on class projects and our research projects. At the same time, he has been a great friend; having me over for dinner, chatting about our (many) common interests, and generally being so approachable.

- My senior colleagues—Ryan Amos, Ben Burgess, Paul Ellenbogen, Arunesh Mathur, and Laura Roberts. Thank you for welcoming me to the CITP community and for always being there to answer my questions.

- Two phenomenal—yet down-to-earth—undergraduate students of whom I had the honor of advising, Kai Ji Kevin Feng and Sten Sjöberg. I am incredibly proud of their transformations into distinguished computer science researchers, and I thank them for the positive influence they have had on my own growth. Both have continued to produce impactful work beyond their time at Princeton; Kevin as a Ph.D. student at a world-class institution and Sten as a security product manager at one of the most prestigious technology companies.

- Jean Butcher and Laura Cummings-Abdo, for always being there for me. Whether it was chit-chat, snacks, reimbursement requests, or a pep talk, I could always rely on both of them to get me through the day.

- All who were part of the CITP community during my time there from 2018 to 2022.

- Ripple for supporting the entirety of my research through a grant from the Ripple University Blockchain Research Initiative.

- Nicki Mahler for always helping with my (many) administrative questions.

- Robert "Bob" Dondero for our time together during all of my teaching assignments. It was an honor to be a teaching assistant in his software engineering course.

- Alison McGregor from the McGraw Center for Teaching and Learning, for working with me over the course of year to improve my public speaking skills.

- My undergraduate research advisor, Andrew Miller, for introducing me to information security research and inspiring me to continue to graduate school. Thank you for taking a chance and believing in me from the very start.

- My friends made at Princeton: Tsung-Lin Hsieh and Tony Ye.

- My childhood, high school, and college friends—Daniel Su, Gabrielle Chen, Peter Louie, Daniel Yi-chia Huang, Amanda Kim, and Yuguang Lingle Lin—for showering me with emotional support, care packages, and food throughout graduate school.

- My parents, my sister, and my dog Pongo, for their unconditional love.

To my parents.

Contents

List of Tables

List of Figures

Bibliographic Notes

Chapter 3 was originally published in 2020:

- Kevin Lee, Benjamin Kaiser, Jonathan Mayer, and Arvind Narayanan. "An Empirical Study of Wireless Carrier Authentication for SIM Swaps". In: *Proceedings of the 16th Symposium On Usable Privacy and Security (SOUPS)*. ed. by Joe Calandrino and Michelle Mazurek. Virtual Conference: USENIX Association, August 2020, pp. 61–79. URL: `https://www.usenix.org/system/files/soups2020-lee.pdf` (visited on 05/20/2022)

Chapter 4 was originally published in 2021. The publication received the best student paper award at the 2021 APWG Symposium on Electronic Crime Research (eCrime):

- Kevin Lee and Arvind Narayanan. "Security and Privacy Risks of Number Recycling at Mobile Carriers in the United States". In: *Proceedings of the 2021 APWG Symposium on Electronic Crime Research (eCrime)*. Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 1–17. DOI: `10.1109/eCrime54498.2021.9738792`

Chapter 5 will be published later in 2022:

- Kevin Lee, Sten Sjöberg, and Arvind Narayanan. "Password policies of most top websites fail to conform to best practices". In: *Proceedings of the Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*. USENIX Association, 2022

While unrelated to my thesis, I am honored to have co-authored the following publications during my PhD:

- Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin. "An Empirical Analysis of Traceability in the Monero Blockchain". In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018), pp. 143–163

- Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. "Blocksci: Design and applications of a blockchain analysis platform". In: *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*. Ed. by Srdjan Capkun and Franziska Roesner. Virtual Conference: USENIX Association, August 2020, pp. 2721–2738. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/kalodner (visited on 05/20/2022)

- Ben Kaiser, Jerry Wei, Eli Lucherini, Kevin Lee, J. Nathan Matias, and Jonathan Mayer. "Adapting Security Warnings to Counter Online Disinformation". In: *Proceedings of the 30th USENIX Security Symposium (USENIX Security)*. Ed. by Rachel Greenstadt and Michael Bailey. Virtual Conference: USENIX Association, August 2021, pp. 1163–1180. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/kaiser (visited on 05/20/2022)

1

Introduction

1.1. Overview

User authentication has made extensive inroads since the 1990s. Just as digital advances across industry have made it more convenient for people to access their bank accounts, health records, and social circles, they have also greatly increased the possibility of unauthorized access by cybercriminals. To defend against this, usable security researchers and standards organizations have prioritized strengthening user authentication. Over the past two decades, they have developed new mechanisms for people to verify their identity (e.g., using SMS, email, smartphone applications, and hardware keys) and have also put forth best practices for companies to follow (e.g., NIST's *Digital Identity Guidelines*) [7–10].

Although the research has flourished, user authentication practice has lagged behind. There are large corporations—even entire industry sectors—that fail to follow up-to-date guidelines, rely on insecure authentication methods, or even inconsistently enforce their authentication policy. At the same time, research has not paid close attention to these practices. From ignoring practical constraints in proposed technical solutions to disincentivizing studies of security policies, the research community has made its progress with marginal societal impact.

We often see the consequences of this disconnect when users become victims of account compromise, payment fraud, and online harassment [11–13]. Most of the time, these anecdotes are disregarded as infrequent occurrences caused by the victim's poor security hygiene, even if they are instead caused by systematic flaws in companies' authentication policies. As a result, these weaknesses persist undiscovered, along with the gap between authentication theory and authentication in the real world.

In this dissertation, we bring to light previously unknown—yet widely-used—user authentication practices. In three studies, we revealed and measured exploits of weaknesses in these practices. These straightforward vulnerabilities have significant societal impact, affecting millions of people. Meanwhile, our outreach from the research has placed pressure on companies to improve their user authentication practices and has even motivated ongoing regulation. In the following sections, we describe the studies:

1.1.1. Customer authentication for SIM swap requests

We examined the authentication procedures used by five prepaid wireless carriers when a customer attempted to change their SIM card. These procedures are an important line of defense against attackers who seek to hijack victims' phone numbers by posing as the victim and calling the carrier to request that service be transferred to a SIM card the attacker possesses. We found that all five carriers used insecure authentication challenges that could be easily subverted by attackers. We also found that attackers generally only needed to target the most vulnerable authentication challenges, because the rest could be bypassed. Authentication of SIM swap requests presents a classic usability-security trade-off, with carriers underemphasizing security. In an anecdotal evaluation of postpaid accounts at three carriers, presented in § A.1, we also found—very tentatively—that some carriers may have implemented stronger authentication for postpaid accounts than for prepaid accounts.

To quantify the downstream effects of these vulnerabilities, we reverse-engineered the authentication policies of over 140 websites that offer phone-based authentication. We rated the level of vulnerability of users of each website to a SIM swap attack, and have released our findings as an annotated dataset on `issms2fasecure.com`. Notably, we found 17 websites on which user accounts can be compromised based on a SIM swap alone, i.e., without a password compromise. We encountered failures in vulnerability disclosure processes that resulted in these vulnerabilities remaining unfixed by nine of the 17 companies despite our responsible disclosure. Finally, we analyzed enterprise 2FA solutions from three vendors, finding that two of them give users inadequate control over the security-usability tradeoff.

This study was done in collaboration with Ben Kaiser, and co-advised by Jonathan Mayer and Arvind Narayanan. Our paper was published in the proceedings of the 2020 Symposium on Usable Privacy and Security (SOUPS) [1].

1.1.2. Security and privacy risks of phone number recycling in the U.S.

We examined the security and privacy risks of phone number recycling in the United States. We sampled 259 phone numbers available to new subscribers at two major carriers, and found that 171 of them were tied to existing accounts at popular websites, potentially allowing those accounts to be hijacked. Additionally, a majority of available numbers led to hits on people search services, which provide personally identifiable information on previous owners. Furthermore, a significant fraction (100 of 259) of the numbers were linked to leaked login credentials on the web, which could enable account hijackings that defeat SMS-based multi-factor authentication. We also found design weaknesses in carriers' online interfaces and number recycling policies that could facilitate attacks involving number recycling. We close by recommending steps carriers, websites, and subscribers can take to reduce risk.

This study was advised by Arvind Narayanan. Our paper was awarded best student paper and was published in the proceedings of the 2021 Symposium on Electronic Crime Research (APWG eCrime) [2].

1.1.3. Password policies of most top websites fail to conform to best practices

We examined the policies of 120 of the most popular websites for when a user creates a new password for their account. Despite well-established advice that has emerged from the research community, we found that only 13% of websites followed all relevant best practices in their password policies. Specifically, 75% of websites do not stop users from choosing the most common passwords—like "`abc123456`" and "`P@$$w0rd`", while 45% burden users by requiring specific character classes in their passwords for minimal security benefit. We found low adoption of password strength meters—a widely touted intervention to encourage stronger passwords, appearing on only 19% of websites. Even among those sites, we found nearly half misusing them to steer users to include certain character classes, and not for their intended purpose of encouraging freely-constructed strong passwords.

This study was done in collaboration with Sten Sjöberg, and advised by Arvind Narayanan. Our paper will be published in the proceedings of the 2022 Symposium on Usable Privacy and Security (SOUPS) [3].

1.2. Key finding: Gaps between user authentication research and practice

Table 1.1.: In this dissertation, we uncovered gaps between user authentication research and practice. ⟶ ⛓️‍💥 represents knowledge in the research community that is not realized in practice, whereas ⛓️‍💥 ⟵ represents a practice of which the research community is unaware of.

	Research		Practice
SIM swaps (Chapter 3)	The research community has never studied weaknesses in call center authentication procedures at mobile carriers.	⛓️‍💥 ⟵	Mobile carriers use insecure authentication methods to verify callers requesting a SIM swap, which can be easily subverted.
	Researchers have recommended against websites using SMS-based authentication due to other security and privacy issues.	⟶ ⛓️‍💥	Websites continue to rely on SMS, and thus do not stand up well against SIM swaps. On 17 websites studied, user accounts can be compromised based on a SIM swap alone.
Recycled phone numbers (Chapter 4)	**Incomplete understanding of legacy systems.** Researchers looking at SMS-based authentication have not considered a fundamental limitation of telephone numbers—that numbers are finite and hence recycled.	⛓️‍💥 ⟵	Standard industry practice is to reassign disconnected phone numbers to other subscribers in order to conserve numbering resources.
Password policies (Chapter 5)	Research has established best practices for blocking the most common passwords, composition rules, and password strength meters.	⟶ ⛓️‍💥	Most of the 120 most popular websites studied were not following the best practices in their password policies.
	The research community has not made an effort to understand why most websites are not following password policy best practices.	⛓️‍💥 ⟵	*(Hypothesized)* Since most websites are not following best practices, there must be underlying reasons for inaction.

These weaknesses embody a broader issue with user authentication: there is a fundamental disconnect between research and practice. Authentication practice has lagged behind research, but it is also the case that the research community has not paid close attention to the practical constraints that have made these vulnerabilities more likely. In Table 1.1, we list our findings from each of the three studies, which illustrate the research-practice gap in user authentication.

While negative effects can be immediately felt in practice with compromises of user safety, research is also affected by the gap. Best practices that are broadly accepted within the community have not been implemented, minimizing the impact of prior studies. Furthermore, researchers without a complete understanding of practical limitations may produce solutions that exclude certain populations (e.g., poor usability of audio CAPTCHAs for low-vision users), or worse, inadvertently introduce exploitable flaws (e.g., dual-use apps in the context of intimate partner violence) [14, 15].

1.3. Context: Widely-deployed user authentication methods

We studied flaws in practices involving widely-deployed user authentication methods that were not necessarily cutting-edge. We did this in order to maximize the relevance our research would have on user safety today. Call centers commonly authenticate callers looking to access their accounts, such as for phone service, credit cards, health insurance, and utilities. According to a 2021 industry report, however, call center fraud has been a frequent channel for account takeovers; it was the leading cause of account takeovers for financial accounts and the second-leading cause at non-financial accounts [16]. Despite the insecurity, there has never been any academic research on call center authentication practices. SMS-based authentication has been used for over two decades. Today, more secure user authentication methods like dedicated software apps and security keys exist, and their adoption has steadily increased. But SMS-based authentication is still widely used; according to a survey of 1,039 respondents, 85% have used SMS 2FA at least once in 2021, compared with 9.7% for security keys and 44.4% for authenticator apps [17]. Similarly, passwords remain the most common means of authentication. Given how the research community prioritizes novelty, our research focused on an understudied area in information security [18–20].

1.4. Context: UI-bound, low-tech adversaries

None of the weaknesses we discovered required exploitation of software bugs. Attackers could remain within the functionality of the user interface and use the system with the same privileges as any other user, albeit with malicious intent. We describe these low-tech attackers as *UI-bound adversaries*, a term coined by Freed et al. [21].

Since the adversary operates within the functionality of the user interface and does not need to use any tools or exploit a system vulnerability, the population of potential attackers is expansive. Impersonating subscribers over the phone to request a SIM swap, intercepting SMS 2FA codes to hijack accounts, looking for vulnerable recycled numbers on number change interfaces, and guessing the password on online accounts are all examples of UI-bound adversaries.[1]

[1] In our study of password policies, we considered both online guessing attacks—which are UI-bound—and offline guessing attacks—which are not UI-bound.

1.5. Context: Policy problems

In these studies, we uncovered weaknesses in different user authentication practices: mobile carriers' call center authentication for SIM swap requests, SMS-based authentication for online services, and password policies of the most popular websites. These weaknesses were caused by flawed security policies, which in turn led to practices that adversely affected the security of user accounts. In Table 1.2, we list the practices we found that were caused by policy problems.

Table 1.2.: In this dissertation, we uncovered weaknesses in different user authentication practices stemming from policy and process flaws.

	Practice	Aspect
SIM swaps (Chapter 3)	All prepaid carriers studied used easily-obtainable information to authenticate callers.	Policy flaw
	Customer service representatives (CSRs) sometimes forgot to authenticate us, proceeded despite us failing authentication, or guided our guesses.	Process flaw
	17 / 145 websites studied allowed SMS 2FA / recovery simultaneously.	Policy flaw
Recycled phone numbers (Chapter 4)	Most number change interfaces had design weaknesses that could facilitate number recycling attacks.	Policy flaw
	CSRs gave inconsistent responses about their carriers' number recycling policy.	Process flaw
Password policies (Chapter 5)	Most of the 120 most popular websites studied were not following the best practices in their password policies.	Policy flaw

Our focus was to examine security policies, but we also found flawed processes. In these instances, the policies were established but were inconsistently enforced due to human error. For example, we found CSRs who weakly enforced the customer authentication policy for SIM swap requests. We also received varied responses when we called CSRs to ask about their carriers' phone number recycling policy, which in turn could lead to subscribers losing their phone numbers and forgetting to update their linked accounts.

None of our findings were the result of software flaws, however. This is significant because of our assumption of a UI-bound adversary; rather than exploit a software bug, they could just exploit the policy and process flaws in place. These exploits are examples of unsophisticated attack vectors that are not considered interesting by the research community, yet have the potential to cause widespread harm.

1.6. Approach: reverse engineering and measurement

Since these practices were not well-documented, we relied heavily on reverse-engineering the security policies at companies. We did so by interacting with their interfaces—just as a normal user would—and documenting any information asked of us. In each of these studies, we overcame unique challenges in order to properly reverse-engineer security policies. In our study of SIM swaps, we followed a call script to ensure consistency between calls, but we also had to adapt to any unexpected circumstances that arose in order to reach our objective: successfully requesting a SIM swap. For instance, when some CSRs questioned us for being unable to verify personal information on our account, we explained we had misentered the information during account setup and apologized for our carelessness. In studying whether websites that offered SMS-based authentication were resilient against SIM swaps, we had to repeatedly enroll in different 2FA / recovery pairs on our test accounts in order to learn the authentication policies, which were often not explicitly stated. In our study of recycled phone numbers, we relied on attempting multiple changes on our accounts, looking for carrier-published FAQs, and inspecting webpage elements when looking for throttling at number change interfaces. Additionally, when we were unable to find any mention of the carriers' number recycling policies, we attempted to reverse-engineer them by making multiple calls to customer service. In our study of password policies, we encountered password change interfaces with minimal or dynamically-shown password advice. We reverse-engineered the password composition requirements to the best of our ability by inputting strings with different character-classes and length, while limiting our combinations to what could be typed on a standard U.S. keyboard. Even with the challenges, reverse-engineering security policies exemplifies the importance of measurement research; it allowed us to discover poor authentication practices that could be easily exploited.

We quantified the prevalence of weaknesses by making multiple measurements across different companies. For example, we reversed-engineered the customer authentication policy for SIM swaps at five U.S. prepaid carriers—all three major carriers and two virtual carriers. In our study of password policies, we measured 120 top websites and found only 13% were following best practices in their password policy. We also made multiple measurements in order to make statistical inferences. In studying recycled phone numbers, we found that UI-bound adversaries could more easily find vulnerable available numbers if they focused their lookups on *Likely recycled* phone numbers, which was statistically significant. We further estimated the inventory of vulnerable available numbers with our study sample, while acknowledging the limitations of our inference with 95% confidence intervals and noting the limited timescale. All in all, we showed that these consequences of the research-practice gap were not one-off occurrences, and thus could be exploited on a regular basis. The scale of these measurement studies has informed users of the risk of harm to their accounts and policymakers looking to learn about the effects of authentication practices at companies.

1.7. Approach: Policy solutions

In our research, we have engaged in outreach with various stakeholders to discuss recommendations: policymakers, carrier trade associations, and online services. We have also publicized our findings for users and journalists to learn about these weaknesses, to make informed decisions on their account security settings, and to educate others. In our study of SIM swaps, we released our findings on carriers and websites after our responsible disclosure. In our study of recycled phone numbers, we filled a knowledge gap in the research community by highlighting a limitation with SMS-based authentication. We plan to publicize the findings of our study on password policies in the coming months.

We have been successful in influencing policy improvements that address these weaknesses. Most notably, in September 2021, the Federal Communications Commission (FCC) launched a formal rulemaking process to protect consumers from SIM swap and number portability attacks, citing our research on SIM swaps as justification [22]. Prior to that, one of the mobile carriers studied had informed us that it had partially implemented our recommendations. Four of the 17 websites that were especially vulnerable to SIM swap attacks informed us that they had fixed the flaws in their authentication policy after our outreach. In light of our research on recycled numbers, the two carriers in our study have made changes to better inform their subscribers about reassigning phone numbers.

All three studies provide a clear pathway for policymakers to learn about current user authentication practices at companies. Policymakers can better understand the severity of these weaknesses from our measurements, and require companies to mitigate the flaws we found. Ultimately, policymakers should intervene when weaknesses lead to harmful effects on users and when the incentives to make fixes are unclear. That is, if responsible disclosure and pressure from users do not lead to improvements to authentication practices, policymakers should require companies to make the improvements or to bear the consequences of noncompliance. Similar to how rulemaking in the 1960s started requiring safety measures in automobiles, government agencies can draw from research to make informed decisions to protect users in the digital space.[2]

1.8. Lessons learned: Studying policies versus studying human subjects

Our goal in all three studies was to reverse-engineer user authentication policy at companies. Some of these efforts required interacting with human systems at large companies (e.g., speaking to CSRs over the phone), but our goal was not to study human behavior.[3] In our SIM swaps study, however, the Princeton University Institutional Review Board (IRB) had identified our protocol as human subjects research, pointing to the CSRs on our calls as the human subjects. While we subsequently complied with the IRB's decision and modified our protocol to reduce risk of harm to CSRs, we also asked the Board to publicly document its stance on corporate policy research, including our "mystery shopper" method. At the time, dozens of IRBs at different institutions had already ruled out corporate policy and "mystery shopper" studies as human subjects research. The lack of guidance has also led to inconsistent decision-making; in our subsequent study of recycled phone numbers, the IRB ruled our protocol as non-human subjects research despite possible human interaction (e.g., communication meant for previous owners of phone numbers studied, communication from previous owners themselves). While

[2] Public Law 89-670.

[3] We make the distinction between small and large companies because small companies may not have pre-established security policies, so interacting with human systems at small companies may require additional consideration.

we appreciate the IRB's guidance in minimizing risks in our protocol, we believe that there should be a distinction between studying corporate policies and studying human subjects. Moreover, official guidance on corporate policy research should be publicly documented, irrespective of the IRB's stance.

1.9. Lessons learned: Ethical computer science research

In this dissertation, we balanced identifying weaknesses in user authentication practices while maintaining research integrity. We embraced principles such as social responsibility, legal compliance, and user safety during all our studies. Although consideration of ethics in technical disciplines has come a long way, designing studies through the lens of established principles and soliciting input from different perspectives is still an unfamiliar exercise to many researchers [23].

In studying SIM swap attacks, we worked closely with the IRB to determine the best way to reverse-engineer customer authentication policies at mobile carriers (§ A.2). On the IRB's suggestion, we did not collect identifying information on CSRs, nor did we release any details about our phone accounts when asked by the carriers studied (since carriers may identify the CSR based on account service records). We also responsibly disclosed our findings at carriers well before publicizing our research. In our study of recycled phone numbers, we took steps to reduce the risk of harm to previous owners of the phone numbers in our study (§ 4.4.1.3, § 4.7.1.4). For instance, we analyzed legitimate calls / texts coming into our phone lines based on metadata online. We did this in order to protect the privacy of previous owners, even though there was no legal issue with reading communication meant for them (we had consulted with legal experts). In our study of password policies, we used leaked passwords from past data breaches, which raises ethical concerns. We believe our use was justified, however, since the passwords were already publicly available and widely used in other password research [24]. Furthermore, we selected leaked passwords to test in order to fully understand websites' defenses, as cybercriminals regularly use these passwords in their attack strategy.

1.10. Lessons learned: Manual measurement methods

In all three studies, we relied on manual measurements instead of running scripts that might have made measurements on a larger scale. In some of these instances, automation was impossible. In order to reverse-engineer authentication policies for SIM swaps, we had to be on the phone with CSRs. Similarly, our human-in-the-loop method to measure SMS 2FA / recovery at websites required us to enroll in out-of-band authentication, such as on our phone or email accounts. At other times, automation required us to account for various UI design patterns in our scripts, and would have still led to restricted results. When we tried to study the password composition policies of top websites, we had piloted a Selenium-based crawler to visit each website's registration page and extract the text displayed near the password field. We encountered many different navigation and registration flows which often prevented our crawler from extracting the entire composition policy (which we spot-checked manually), on top of the fact that several websites had measures in place to prevent automated crawlers. We hence decided to make the measurements by hand rather than augment our scripts each time we encountered a flow that had not been accounted for.

We had additionally tried crowdsourcing our measurements with marginal success. In our study of password policies, we recruited Workers on Amazon Mechanical Turk (MTurk) to visit an assigned website, navigate to the registration page, and reverse-engineer the password composition policy by trying example strings we provided. Even though we paid more to recruit highly-rated Workers, we had poor results. More than half of the websites had disagreeing data points, and a significant portion of responses incorrectly reported that there was no registration page present (presumably in order to finish the Task early).

We ultimately relied on our own manual work to prove the relevance of these weaknesses. In studying SIM swap attacks, we made one phone call per account—50 calls in total (~40 hours of work)—to customer service at the five carriers in order to reverse-engineer their authentication policy. To ensure consistency, we asked our hired research assistants—the callers—to follow a script, and to purposely abstain from providing information that was not part of our threat model. When checking how well websites stood up against SIM swap attacks, we created accounts at 145 websites, provided all requested personal information, and examined the 2FA / recovery option pairs that were available to us. In studying recycled phone numbers, we logged available numbers on the number change interfaces at two major carriers (~50 hours of work). We then checked whether 259 of the *Likely recycled* phone numbers were still linked to existing profiles at six popular websites. In studying password policies at 120 of the most popular websites, we reverse-engineered each website's blocklist strategy, composition rules, and password strength meter behavior. In one analysis, we tested each website's ability to prevent 40 of the most common passwords from being set on our test accounts; we thus attempted 4,800 password changes in that analysis alone (~200-300 hours of work).

Even though our manual methods may have produced fewer data points than that of automated methods, we were able to answer important questions about the authentication practices we studied. We were able to show that these weaknesses were not one-off occurrences, and thus could be exploited on a regular basis (§ 1.6). With our measurements, we provided novel insights into these practices which resulted from flawed policy and processes, while also remaining careful about the statistical inferences we could make with our results.

1.11. Lessons learned: Challenges with reporting policy-related vulnerabilities

We responsibly disclosed any vulnerabilities that could harm users at affected companies. In these disclosures, we demonstrated the feasibility of an attack, the population affected, and our recommendations to reduce the risk of harm. In our study of SIM swaps, we presented our findings to mobile carriers at a meeting with CTIA—the trade association for U.S. mobile carriers. We also presented our findings and recommendations to the 17 websites found to be allowing SMS for 2FA and account recovery simultaneously. In our study of recycled phone numbers, we pointed out flaws in carriers' user interfaces which enabled attacks in another meeting with CTIA.

While we have effected changes with our outreach, we did so in the face of challenges with reporting policy-related vulnerabilities, which were sometimes dismissed as being out-of-scope. That's because companies are used to receiving reports about bugs in their software and infrastructure, as opposed to policy-related vulnerabilities. Most notably, in our disclosure to the 17 websites in the SIM swaps study, five did not understand our vulnerability report despite our attempts to make it as clear as possible; three websites acknowledged SIM swap attacks but failed to realize that their authentication policy was allowing for vulnerable accounts, and the other two websites misrepresented our disclosure as feature requests (§ A.3). Moreover, four websites relied solely on third-party bug bounty platforms to be notified about vulnerabilities. Only one of our attempts through bug bounties was successful, since members of the platform were only focused on reviewing software bugs. In our study of recycled numbers six months later, we studied four of the original 17 vulnerable websites and found all four still allowed simultaneous use of SMS for 2FA and recovery.

All in all, only four websites immediately made changes in response to our disclosure. Through this experience, we found that companies were generally unaware of vulnerabilities in their authentication policy and had no direct contact methods for security reporting procedures [25]. We continue to call for improvements to reporting policy-related vulnerabilities, and remind other security policy researchers of the need to account for these ongoing challenges in their own outreach.

1.12. The need for research that aligns with societal benefit

The studies here are ultimately guided by potential for impact; we studied weaknesses in user authentication practices because of their relevance to user safety. We showed that attacks exploiting these flaws were straightforward. Moreover, these unsophisticated attacks can target millions of people; customers at U.S. mobile carriers, users that rely on SMS-based authentication to secure their online accounts, and users with weak passwords. By discovering these weaknesses, measuring attacks exploiting those weaknesses, and releasing our findings and recommendations, we hope that our work begins to bridge the gap between user authentication research and practice.

Unfortunately, the security research community oftentimes is not interested in studying these vulnerabilities. That's because the current incentives for publication within the community are misaligned with societal impact. Rather, reviewers at top conferences disproportionately merit work for publication based on factors like emerging technologies and sophisticated attack vectors. Studying authentication policies, for instance, has revealed direct flaws in current practices, but has not been nearly as valued by the community. This paints a stark contrast to the kind of technology-enabled harm that happens in reality: most attacks are on current or older technologies, with attackers using the products as intended, albeit with malicious intent.

The predisposition towards novelty and complexity in academia is not a new issue; several established figures have already raised this notion in invited talks and opinion pieces [18, 19]. These biases have prevented rather straightforward vulnerabilities from being both studied and published, including weaknesses in user authentication practice. For researchers, turning a blind eye to studying these vulnerabilities persists the research-practice gap and much of the technology abuse that happens in the world.

The community needs to reorient its focus on how much good their research can do for the world. By moving away from the predisposition towards novelty and complexity and using new factors to determine a study's value, researchers may identify further opportunities for impact that were previously unrealized. They could instead consider factors such as skills needed of an attacker (low-tech adversary v.s. high-tech adversary) and manifestation in the real world (hypothetical attacks v.s. actual attacks with anecdotal evidence in the media or complaints filed at government agencies).[4]

As we show in this dissertation, studying user authentication practices is a prime example of security research that aligns with societal benefit. Through reverse-

[4] In fact, if we create a matrix using those two factors just described, most security work published today falls under the "high-tech adversary x hypothetical attacks" category, which has the least societal impact.

engineering security policies at companies, quantifying the risk of harm through manual measurements, and going beyond the writeup with outreach, researchers can influence public policy to protect users from abuse. They can also give valuable insights back to the community in order to harmonize future research with practice.

1.13. Outline

The dissertation is structured as follows. Chapter 2 provides the necessary background. Chapters 3 to 5 provide the results of the three aforementioned studies. We discuss avenues for future work and conclude in Chapter 6.

2

Background and related work

2.1. User authentication

User authentication is the process of verifying a user's identify. To authenticate, the user typically proves her identity by providing one of the following credentials:

- something she knows (e.g., a password or PIN),
- something she has (e.g., her phone), or
- something she is (e.g., her fingerprint).

The verifier, which enforces the underlying system's user authentication policy, can then use the provided credentials to successfully or unsuccessfully verify the user's identity. Authentication policies are comprised of rules such as types of credentials allowed, number of required credentials, password composition rules, and lockout procedures.

In the digital space, it is important for user authentication to be secure in order to prevent unauthorized access to critical data, as well as usable in order to allow verified users access to their data and the ability to perform actions within the system. With those objectives in mind, researchers have proposed various authentication mechanisms over the years, such as hardware security keys, one-time passcodes (OTPs) sent via telephone, as well as password complexity guidelines. We often interact with these mechanisms when we access our personal devices (e.g., laptops, phones, smart home devices) and online services (e.g., bank accounts, social media accounts, phone accounts). In this dissertation, we focus on user authentication for online services.

2.1.1. Multi-factor authentication

Multi-factor authentication (MFA) is a security enhancement that requires users to provide two or more different types of credentials for authentication. This typically involves entering a password along with a second method, such as an OTP sent via text message, or tapping a notification sent to a registered smartphone. MFA—or 2FA as it is commonly referred to—makes it more difficult for attackers to break into accounts, since compromising one credential is not enough to gain access. Some methods may be less secure than others, however. Moreover, as we discovered through our research, some companies have flaws in their authentication policy that allow 2FA to be bypassed with just a single factor.

2.1.2. Call center authentication

An important context for user authentication is in call centers. Call center authentication is the process of verifying a user's identity over the phone channel in order to access her account; the customer service representative (CSR) on the call acts as the verifier and may present the caller with several authentication challenges, such providing an SMS OTP and answering personal knowledge questions. Call centers provide an additional means of online account access if the web interface is temporarily offline, the user does not have internet access, or the user needs human assistance. Unlike user authentication on a web interface, call center authentication usually involves interacting with a human, who may be susceptible to social engineering attacks, bribery, or human error. In Chapter 3, we studied call center authentication practices at mobile carriers for customers requesting a SIM swap.

2.2. SIM swaps

2.2.1. SIM cards and number portability

Wireless service to a mobile device is tied to that device's SIM card. Wireless carriers keep track of the mapping between phone numbers and SIMs to ensure that calls, messages, and data connections are routed to the correct customer. Generally, the mapping from a phone number to a SIM is a one-to-one relationship: a phone number can only be associated with a single SIM at any given point in time and vice versa.

SIM cards further the bring-your-own-device (BYOD) policy that exists at many carriers today: users are usually free to bring their own devices to the network, provided that the device is not locked to another carrier and that the customer purchases a new SIM card. Similarly, if a user were to ever switch devices, they could easily remove their existing SIM card and insert it into the new device. The customer could also purchase a new inactive SIM card, provide a CSR at the mobile provider with the new card's Integrated Circuit Card Identifier (ICCID), and migrate the service over to the new SIM before inserting it into the new device. From then, service on the original device would be disconnected, and all connections would move over to the new device with the now-activated SIM.

In the U.S., customers also have the option of taking their phone numbers with them whenever they switch carriers; a user seeking to move their number to a new provider would provide their old account details to their new provider, who would in turn request the number from the original provider. After validating the request, the original provider would push their number over to the new carrier. Local number portability—as this is called—is regulated by the Federal Communications Commission (FCC), allowing customers to switch carriers while retaining their original numbers for little to no cost.

There are two scenarios in which an account holder would need to change the SIM card in their device: a SIM swap or a *port out*. In a SIM swap, the account and phone number stay with the original carrier, and only the SIM card is changed. In a port out, the number is transferred to a new account at a new carrier. Both types of account changes involve switching SIM cards; SIM swaps use cards from the same carrier whereas port outs use cards from different carriers.

We studied SIM swaps in chapter 3 due to their relative simplicity; we cannot be confident that the authentication procedures for SIM swaps and port outs are the same. It is worth noting the distinction that SIM swaps typically take no more than two hours (and are often instantaneous), while port outs can take several days.

2.2.2. Related work

SIM swapping is not the only means to intercept calls and SMS messages. There are man-in-the-middle (MITM) attacks that take advantage of weaknesses in mobile phone network infrastructure. For instance, IMSI-catchers [26] can be used to intercept nearby connections on certain older wireless protocols by posing as a mobile tower and forcing phones in the vicinity to connect to it. From there, the IMSI-catcher can force connected phones to use vulnerable encryption or none at all, rendering calls and SMS unprotected. IMSI-catchers take advantage of a weakness in design: legacy cellular networks do not support cell tower authentication. That is, nearby phones are forced to downgrade their connections in order to use legacy cellular network protocols. Though initially used by authorities only, IMSI-catchers can now be built with commercially available components and used by anyone [27].

In Long-Term Evolution (LTE) networks, mobile devices are assigned a Globally Unique Temporary ID (GUTI) in order to alleviate the location-tracking implications of IMSI-catchers. As the name suggests, an temporary identifier is assigned to the device by the access network. The GUTI is then periodically updated to inhibit device tracking. However, as there are no standard guidelines for when and how to update the GUTI, many carriers have been mishandling reallocations either by reusing the same GUTI or assigning predictable identifiers. Shaik et al. showed that repeated calls using Voice over LTE (VoLTE) could reveal a victim's location, since the same GUTI is reallocated [28]. Hong et al. showed that 19 out of 28 carriers across 11 countries were reallocating GUTIs in predictable ways; reallocated GUTIs contained patterns that could be linked back to the previous ones [29]. They also proposed a scalable unpredictable GUTI reallocation mechanism. There are also weaknesses in the framework that enables carrier interoperability, namely the Signaling System 7 (SS7) protocol, which is designed to trust all requests. The weaknesses of SS7 have long been documented [30]; in 2014, researchers discovered how SMS can be intercepted using the SS7 protocol [31, 32]. Recently, criminals used an SS7 attack to intercept SMS MFA messages for bank accounts, resulting in financial loss [33].

SS7 has been replaced with Diameter—an improved signaling protocol that supports encrypted requests—with the rollout of 4G and 5G networks, but there are still many carriers in the network that do not use authentication, leading researchers to discover new Diameter-based SMS attacks [34].

While IMSI-catchers and SS7 attacks represent significant threats to the security of mobile communications, SIM swap attacks are inexpensive, low-risk, and as we show, very effective for account hijacking attacks. This makes them attractive to a host of adversaries, including those for whom IMSI-catchers and SS7 attacks are out of reach. Thus, our study focuses on this urgent threat.

There has also been research on customer authentication in other industries. Bonneau et al. examined the use of personal knowledge questions at Google; they discovered that a significant portion of users (37%) provided false answers in order to make them “harder to guess” [35]. Personal knowledge questions among English-speaking users had low rates (60%) of success, as most users could not recall their answers when asked. Colnago et al. [36] observed the deployment of a software token 2FA system at Carnegie Mellon University, and found that while adopters found 2FA annoying, they found it fairly easy to use. The study also found that adopters who were forced to enroll in 2FA had a slightly negative perception of it, as opposed to adopters who were offered to enroll. Weir et al. examined user perceptions of security and usability in online banking, and found that nearly two-thirds of participants chose the device they perceived least secure (but most convenient) as their preference [37]. Redmiles et al. empirically examined the relationship between the proportion of users signing up for SMS-based 2FA based on perceived risk [38]. In the study, users of a testbed bank website were informed of the risks of account hackings and offered to enroll in SMS-based 2FA. Accounts were then randomly selected on a daily basis to be “hacked”, weighted by their 2FA settings. The study found that participants were more likely to make these decisions when faced with higher risk.

2.3. Phone number recycling in the United States

2.3.1. The North American Numbering Plan

In the United States, telephone numbers are formatted and geographically assigned according to the North American Numbering Plan (NANP). Developed by the Bell System (later known as AT&T) in the 1940s to unify inconsistent and unorganized numbering across its various regional telephone networks, the NANP has expanded to comprise the Public Switched Telephone Network (PSTN) in 20 North American countries and their territories. This has served to reduce long-distance international dialing confusion within the NANP network: all numbers are fixed-length and all countries utilize the same international calling code ("1").

The North American Numbering Plan Administrator (NANPA) serves as the supervising body for all NANP resources. As a neutral entity, the NANPA oversees interactions between NANP member countries, including disputes, audits, requests, and most importantly, number allocation. Each participating country maintains a regulatory authority over its assigned numbering resources. In the U.S., the Federal Communications Commission (FCC) serves as the regulator for U.S.-assigned phone numbers. Additionally, the FCC serves a plenary role: it periodically appoints a new administrator from the private sector to serve the position. At the time of writing, Somos, Inc. is serving as the NANPA under a five-year contract.

All NANP phone numbers are of the 10-digit format:

NPA-NXX-XXXX

- The number plan area (NPA) code, or area code, comprises the first three digits. The first digit can be in range [2,9], while the second and third digits can be in range [0,9].

- The central office (exchange) code (NXX) comprises the the next three digits. The first digit can be in range [2,9], while the second and third digits can be in range [0,9].
- The line number (XXXX) comprises the last four digits of the telephone number. All digits can be in range [0,9].

The NANP divides all territory into distinct NPAs, and assigns a three-digit area code to each region. New area codes are primarily added through NPA splits or NPA overlays. In an NPA split, the original NPA is partitioned into two smaller NPAs; one keeps the original area code, while the other is assigned the new area code. All customers in the NPA with the new area code would have their numbers replaced with new ones, freeing up resources in the original area code. In an overlay, a new area code is additionally assigned to one or more adjacent NPAs. Existing customers keep their numbers, but new customers may be assigned numbers with the new overlay code. In New York City, area code 212 was split to only cover Manhattan in 1984, customers in the other boroughs were assigned the new 718 area code. In 1999, area code 347 was added as an overlay for 718. As of May 2022, there are currently 343 area codes in use in the U.S.

2.3.2. Numbering resources and exhaustion

Historically, all carriers looking to set up service in a region were assigned an exclusive NXX within the corresponding area code, that is, blocks of 10,000 contiguous numbers. Upon the advent of new technologies—cable modems and Voice over IP (VoIP), coupled with the Telecommunications Act of 1996, barriers-to-entry were lowered, and many new local carriers sprung up in a suddenly competitive environment. As a result, available NXX assignments were rapidly depleted and new area codes had to be deployed, leading the director of the then NANPA to speculate that 10-digit phone numbers would be completely exhausted by 2025, thereby capping expansion [39]. The proliferation of new and unfamiliar area codes also contributed to the severity of the 809 scam—a social engineering attack that baits U.S. subscribers into returning missed calls to premium-rate numbers in the Caribbean.

In 2000—in an effort to combat number hoarding and resource exhaustion—the FCC reassigned the authority of reclaiming unused NXXs to the states, away from the NANPA. State commissions could now investigate whether NXXs were being activated (made available to subscribers) within six months of assignment to the carrier, and order the NANPA to reclaim the resources otherwise. In 2001, the FCC introduced thousands-block number pooling (or simply, number pooling)—the allocation of 1,000 number blocks (NXX-X) to carriers. This essentially allowed carriers in the same service region to use the same NPA-NXX, reducing the amount of unused numbers and the rate of exhaustion. With the rollout of number pooling, carriers with entire NXX blocks in certain jurisdictions were required to donate unused to lightly used NXX-Xs back to NANPA. Carriers would also have to prove that they have less than a six-month inventory remaining in the service area before requesting additional numbers. Number pooling is currently mandatory in the top 100 Metropolitan Statistical Areas (MSAs) and in states that require number pooling, but it remains optional in most of the U.S.

Recent NANPA estimates from October 2020 predict that 10-digit phone numbers will be exhausted by 2050 [40].

2.3.3. Number recycling

The NANPA only activates new area and central office codes when absolutely necessary. With the FCC-imposed restrictions on NANP resources in the U.S., carriers must also strategize and plan their number assignments efficiently. To satisfy inventory and utilization requirements, carriers may choose to return disconnected blocks or reassign them to other customers. Carriers routinely pursue the second option by placing numbers back into their pool upon disconnection of service and making them available for reassignment after a waiting period. According to the FCC, 35 million phone numbers are disconnected and placed back in the pool every year [41]. As a result, new subscribers who select "new" numbers will often end up receiving communication meant for the previous owners, from threatening robocalls to personal texts.

2.3.4. Related legislation

Under FCC rules, all telecommunications carriers that receive U.S. numbering resources are required to semi-annually report resource and utilization statistics, unless mandated otherwise by state commissions.[1] Carriers are also limited to a six-month inventory of telephone numbers in each of their service areas.[2] With regards to number recycling, carriers are prohibited from reassigning disconnected numbers until 45 days have elapsed since disconnection, and can age numbers for up to 90 days (365 days for numbers assigned to business customers).[3]

The FCC has taken interest in phone number recycling by way of combating unlawful robocalls made to reassigned numbers. Specifically, previous owners of recycled numbers may have consented to robocalls, whereas current owners may find such calls undesirable, but may not be given a chance to consent. Under the Telephone Consumer Protection Act of 1991 (TCPA), certain telephone calls—such as robocalls—made without the called party's consent are prohibited. In December

[1] 47 C.F.R. § 52.15(f)
[2] 47 C.F.R. § 52.15(g)(4)(iii)
[3] 47 C.F.R. § 52.15(f)(1)(ii)

2018, the FCC announced a plan to create a reassigned number database, along with establishing the 45-day minimum aging period [42]. Carriers would be mandated to report recycled numbers on a monthly basis, which would be compiled into a centralized source. Callers can then check for reassigned numbers against their calling lists before initiating communication, thereby reducing the possibility of TCPA violations from calling new subscribers.

In December 2020, the commission selected SomosGov—a wholly-owned subsidiary of Somos, Inc. (the current NANPA)—as the Reassigned Numbers Database Administrator (RNDA) [43]. In November 2021, the RND became operational to FCC-verified accounts for a fee.

2.4. Passwords

2.4.1. Password guessing attacks

Password guessing is a procedure used by attackers to learn a user's password and subsequently break into her account. There are two classes of password guessing attacks: online and offline attacks. In an online attack, the adversary makes repeated attempts to log into the victim's online account, varying the password between each try. Websites may rate-limit login attempts, hence the number of guesses an adversary can make is far lower than in an offline attack. In an offline attack, the adversary obtains a database of stolen (usually obfuscated) passwords and tries to recover the passwords through large scale automated guessing scripts. Offline attacks are constrained by factors such as hardware, guessing algorithm, and the database hash (obfuscation) method, but can generally make more guesses than online attacks.

In both cases, adversaries usually make informed guesses to avoid wasting resources, whether it is to avoid the rate limit at websites (online attacks) or to save on computing power (offline attacks). To that end, they can vary the guessing algorithm as well as the wordlists used to train the algorithm. Most wordlists comprise of dictionary words and lists of common passwords. The research community has extensively studied the algorithms, wordlists, and economics of password guessing attacks [24, 44, 45].

2.4.2. Modeling password strength through adversarial guessability

Password strength has traditionally been measured using Shannon entropy, a function of the counts of lower- and uppercase letters, digits, and symbols (LUDS). While previously recommended by the National Institute of Standards and Technology (NIST), entropy—also commonly referred to as complexity—turned out to be a poor proxy for password security [46]. Researchers soon found mismatches between password entropy scores and time needed for attackers to crack a password (or a set of passwords) [45]. The information security community has since favored using guessability as a measure of password security [44, 47].

Guessability more closely resembles the practically important sense of password strength: the actual number of guesses an adversary would require to correctly guess the password. Unlike Shannon entropy, guess number metrics can factor in contextual information such as common passwords, human predictability and composition rules presented at password creation [45]. However, the attack method and configuration matters: many previous studies—facing time and resource constraints—have only been able to model specific attackers by using only one attack method with limited training data. A necessary drawback to the guessability approach is its inherent subjectivity. Whereas entropy is an objective measure, there is no objective guess number for any password; adversarial guessing is a strategic problem and different strategies will produce different results over the same password set input. For this reason, comparisons between studies using different guessing algorithms can be difficult at best, and moot at worst.

In an effort to harmonize future studies, in 2015, the Passwords Research Team at Carnegie Mellon University released Password Guessability Service (PGS)—a free service that rates the strength of submitted passwords [24]. PGS simulates a real attacker guessing passwords; it leverages multiple (5 at the time of our study) cracking tools to arrive at the user-provided plaintext password. Using each tool, PGS calculates the guessability (i.e., the guess number) as the password's strength rating. PGS also offers the `min_auto` configuration, which returns the minimum guess number for each password across all 5 tools. Previous research has found that the `min_auto` approach provides a conservative estimate for the performance of an unconstrained professional attacker [24].

The research community generally considers 10^6 guesses to be the upper limit of online guessing attacks, and currently considers 10^{14} guesses to be the upper limit for offline guessing attacks.

3

An Empirical Study of Wireless Carrier Authentication for SIM Swaps

3.1. Introduction

Mobile devices serve many purposes: communication, productivity, entertainment, and much more. In recent years, they have also come to be used for personal identity verification, especially by online services. This method involves sending a single-use passcode to a user's phone via an SMS text message or phone call, then prompting the user to provide that passcode at the point of authentication. Phone-based passcodes are frequently used as one of the authentication factors in a multi-factor authentication (MFA) scheme and as an account recovery mechanism.

To hijack accounts that are protected by phone-based passcode authentication, attackers attempt to intercept these passcodes. This can be done in a number of ways, including surveilling the target's mobile device or stealing the passcode with a phishing attack, but the most widely reported method for intercepting phone-based authentication passcodes is a SIM swap attack. By making an unauthorized change to the victim's mobile carrier account, the attacker diverts service, including calls and messages, to a new SIM card and device that they control.

SIM swap attacks allow attackers to intercept calls and messages, impersonate victims, and perform denial-of-service (DoS) attacks. They have been widely used to hack into social media accounts, steal cryptocurrencies, and break into bank accounts [48–50]. This vulnerability is severe and widely known; since 2016 NIST has distinguished SMS-based authentication from other out-of-band authentication methods due to heightened security risks including "SIM change" [51].

SIM swap procedures have valid purposes: for example, if a user has misplaced their original device or acquired a new device that uses a different size SIM card slot than the device it is replacing. In these cases, customers contact their carrier (often by calling the carriers' customer service line) to request a SIM card update on their account. The customer is then typically presented with a series of challenges that are used to authenticate them. If the customer is successfully authenticated, the customer service representative (CSR) proceeds to update the SIM card on the account as requested.

We examined the types of authentication mechanisms in place for such requests at five U.S. prepaid carriers—–AT&T, T-Mobile, Tracfone, US Mobile, and Verizon Wireless—–by signing up for 50 prepaid accounts (10 with each carrier) and subsequently calling in to request a SIM swap on each account.[1] Our key finding is that, at the time of our data collection, all five carriers used insecure authentication challenges that could easily be subverted by attackers. We also found that in general, callers only needed to successfully respond to one challenge in order to authenticate, even if they had failed numerous prior challenges in the call. Within each carrier, procedures were generally consistent, although on nine occasions across two carriers, CSRs either did not authenticate the caller or leaked account information prior to authentication. These findings are consistent with a policy that overemphasizes usability at the expense of security.

[1] Unlike a postpaid account, registering a prepaid account does not require a credit check, making it easy for one researcher to sign up for multiple accounts. Authentication procedures may differ for postpaid accounts.

Our testing results offer insight into the security policies at major U.S. prepaid mobile carriers with implications for the personal security of the millions of U.S.-based customers they serve. We also offer recommendations for carriers and regulators to mitigate the risks of SIM swap attacks.

Next, we evaluated the authentication policies of over 140 online services that offer phone-based authentication to determine how they stand up to an attacker who has compromised a user's phone number via a SIM swap. Our key finding is that 17 websites across different industries have implemented authentication policies with logic flaws that would enable an attacker to fully compromise an account with just a SIM swap.

Finally, we analyzed enterprise MFA apps offered by Duo Security, Okta, and Microsoft, to further understand the downstream impact of SIM swaps. Our finding is that Duo enables SMS-based MFA by default (and makes it difficult to disable), which introduces security risks. The default authentication policies at Duo and Okta sit on opposite ends of the security-usability tradeoff, with Duo overemphasizing usability by default and Okta overemphasizing security.

Responsible disclosure and responses. In July 2019 we provided an initial notification of our findings to the carriers we studied and to CTIA, the U.S. trade association representing the wireless communications industry. In January 2020, T-Mobile informed us that after reviewing our research, it had discontinued the use of call logs for customer authentication.[2]

We reported our MFA configuration findings to the 17 vulnerable websites in January 2020 (§ 3.7.3). We document the widespread failures we encountered in the vulnerability disclosure processes established by companies, including the fact that

[2] Some carriers asked the customer for information that can be obtained from call logs for authentication, such as the phone number of the last placed or received call. The use of call logs—whether incoming or outgoing—for authentication is insecure because attackers can call the victim or trick the victim into placing a call.

many companies have no process to report security policy vulnerabilities as opposed to software bugs. As a consequence, nine of the 17 websites remain vulnerable, which cumulatively have billions of users.

3.2. Background

Carrying out an unauthorized SIM swap or port out to hijack a victim's phone number is obviously unlawful—at minimum a violation of the Computer Fraud and Abuse Act (CFAA) and possibly wire fraud or wiretapping. Authorities and companies have posted advisories against using SMS for two-factor authentication (2FA), most notably in 2016 when the National Institute of Standards and Technology (NIST) initially declared SMS-based authentication to be deprecated in its draft of *Digital Identity Guidelines* [51]. NIST slightly softened its stance a year later by categorizing SMS-based authentication as "restricted"—an authentication factor option that carries known risks [7]. The rise in SIM swap scams has recently led organizations like the Better Business Bureau (BBB) to issue warnings to consumers against using their phone numbers for authentication [52].

3.2.1. Phone-based authentication

Phone-based passcodes are a common authentication technique. They are typically used as one of multiple authentication factors, as a backup authentication option, or as an account recovery method. A passcode can be transmitted to a user's phone via an SMS text message, a phone call, an email, or an authenticator app. The Internet Engineering Task Force (IETF) has published standards for generating, exchanging, and verifying passcodes as part of an authentication procedure [9, 10].

We distinguish passcodes delivered by SMS and phone calls from the other phone-based passcode authentication methods (authenticator apps and email passcodes). The former are susceptible to SIM swap and port out vulnerabilities because they are tied to a phone number and the associated cellular service; the latter are not. In the balance of the chapter, we consider only passcode authentication via SMS and phone call and use the terms "SMS-based authentication" and "SMS-based 2FA" to describe these methods.

3.3. Threat model

We assumed a weak threat model: our simulated attacker knew only information about the victim that would be easily accessible without overcoming any other security measures. Specifically, our attacker knew the victim's name and phone number. We also assumed that the attacker was capable of interacting with the carrier only through its ordinary customer service and account refill interfaces, and for purposes of one attack, that the attacker could bait the victim into making telephone calls to a chosen number. Other than providing scripted answers and persisting through failed authentication challenges, the research assistants (RAs) simulating our attacker used no social engineering tactics. As we will show later, this weak attacker was able to defeat several different authentication challenges used by carriers.

We note that many realistic adversaries could gain access to additional information that could be used to bypass challenges. They could also seem more credible by spoofing the victim's caller ID or escalating the request to management, none of which were included in our method. By assuming such a conservative threat model, we provide a lower bound on real-world attacker success rates.

3.4. Method

The goal of a SIM swap attack is to convince the carrier to update the SIM card associated with a victim's account, thereby diverting service from the victim's SIM and phone to a new SIM and phone in the adversary's possession.

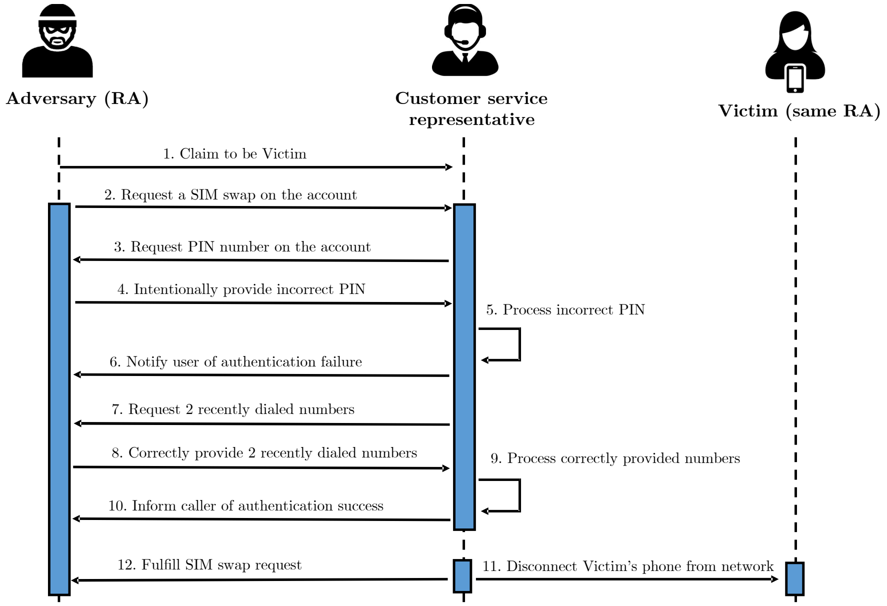

Adversary (RA)
Customer service representative
Victim (same RA)
1. Claim to be Victim
2. Request a SIM swap on the account
3. Request PIN number on the account
4. Intentionally provide incorrect PIN
5. Process incorrect PIN
6. Notify user of authentication failure
7. Request 2 recently dialed numbers
8. Correctly provide 2 recently dialed numbers
9. Process correctly provided numbers
10. Inform caller of authentication success
12. Fulfill SIM swap request
11. Disconnect Victim's phone from network

Figure 3.1.: An example scenario from following our call script. The adversary (the research assistant) intentionally fails the first authentication scheme, but correctly answers the second one because of its inclusion in the threat model. The victim (the same research assistant) receives a notification about an account change when the SIM swap is complete.

In our study, we sought to reverse-engineer the policies for SIM swaps at five U.S. carriers—AT&T, T-Mobile, Tracfone, US Mobile, and Verizon Wireless. We answer the following questions:

1. What are the authentication procedures that prepaid carriers use for SIM swaps? Are they consistent within carriers? Are they consistent across carriers?
2. Do SIM swap authentication procedures withstand attack?

3. What information would an attacker need about their victim to perform a SIM swap attack? Can the attack be perpetrated using only easily acquirable information?

Tracfone and US Mobile are mobile virtual network operators (MVNOs), meaning that they do not own their own wireless network infrastructure and instead contract access to the infrastructure of other networks. The MVNO marketplace is diverse: there are dozens of companies in the U.S. serving a combined subscriber base of over 36 million. Tracfone is a 20-year-old company that currently services over 25 million customers; US Mobile is a much smaller and newer provider, founded in 2014 and serving just 50,000. The difference in their age could suggest different policies for authenticating customers, so we included them in our study.

We created 10 simulated identities for our study and assigned each a name, date of birth, geographical location, and email address. For each identity, we registered prepaid accounts at all five carriers, using SIM cards we had purchased from electronics stores. The accounts were funded with prepaid refill cards purchased at local retail outlets; in a few cases we used one-time virtual debit cards instead. Due to the possibility that carriers log seen phones, we did not reuse devices between experiments; that is, each identity was assigned a unique "victim phone" and "adversary phone," for a total of 20 devices. For each account, we spent at least a week making and receiving phone calls and text messages to generate usage history. At the end of this phase, we hired research assistants (RAs)—who had been designated as the account owners at signup—to call the customer service number for the carrier and request that the SIM card on the account be updated to a new SIM card in our possession. We placed each call from a device that was not registered to the account being studied. During the call, we took notes on what pieces of authenticating information the CSR requested and whether or not the swap was ultimately successful. We did not record or transcribe the calls.

On the calls, all RAs followed the same script: they informed the CSR that their SIM appeared to be faulty because service on the device was intermittent, but that they had a new SIM card in their possession they could try to use. They then responded to any authentication challenges the CSR posed. If the RA could not answer an authentication challenge correctly within the capabilities of the simulated attacker (see § 3.3), the RA was instructed to claim to have forgotten the information or to provide incorrect answers. When providing incorrect answers to personal questions such as date of birth or billing ZIP code, RAs would explain that they had been careless at signup, possibly having provided incorrect information, and could not recall the information they had used. An example scenario from following our call script is shown in Fig. 3.1.

If the SIM swap was successful, we inserted the new SIM into a different device—the "adversary-controlled phone"—and proceeded to make a test call. We also made a test call on the original device to ensure that cell service had been successfully diverted. If the CSR had insisted on remaining on the line until the swap was completed, we gave a verbal confirmation and then ended the call. The experiments ran from May through July of 2019.

In all cases, the same RA simulated both the attacker and the victim, so there were no unauthorized transfers. The accounts were at all times controlled by the research team. RAs were paid standard institutional RA rates. While the purpose of the study was to understand carrier policies and practices, out of an abundance of caution we sought and obtained approval from Princeton University's Institutional Review Board. We provide additional details about mitigating risks in our study in § A.2.

Our initial IRB application was submitted and approved in March of 2019 and April of 2019, respectively. We provided initial notification to the carriers we studied and CTIA on July 25, 2019. We presented our findings in-person to major carriers and CTIA in September 2019.

3.5. Results

We documented how the mobile carriers we studied authenticate prepaid customers who make SIM swap requests. We observed providers using the following authentication challenges:

- **Personal information**: street address, email address, date of birth
- **Account information**: last 4 digits of payment card number, activation date, last payment date and amount
- **Device information**: IMEI (device serial number), ICCID (SIM serial number)
- **Usage information**: recent numbers called (call log)
- **Knowledge**: PIN or password, answers to security questions
- **Possession**: SMS one-time passcode, email one-time passcode

Table 3.1.: Authentication methods that we observed at each carrier. A checkmark means that a type of information was a component of at least one pathway for SIM swap customer authentication; it does not mean that a type of information was necessary or by itself sufficient for SIM swap customer authentication.

	Personal Information			Account Information			Device Information		Usage Information	Knowledge		Possession	
	Street Address	Email Address	DOB	Last 4 of CC	Activation Date	Last Payment	IMEI	ICCID	Recent Numbers	PIN or Password	Security Questions	SMS OTP*	Email OTP
AT&T					●	●	●	●	●	●		●	
T-Mobile									●	●		●	●
Tracfone	●	●	●				●	●		●	●	●	
US Mobile	●	●		●				●					
Verizon						●	●	●	●	●		●	

🟩 generally accepted in the computer security research field

🟥 had not been previously tested but we demonstrate is insecure (for reasons explained below)

🟨 known to have security shortcomings (also for reasons described below)

Table 3.1 presents the authentication methods that we observed at each carrier. Green represents secure authentication methods, yellow fields contain methods with known vulnerabilities, and red represents authentication methods that had not been previously documented and that we demonstrated are insecure. A circle in a cell indicates that on at least one call to the carrier's customer service, while attempting a SIM swap, a CSR requested that information to authenticate the subscriber. In other words, a checkmark means that a type of information was a component of at least one pathway for SIM swap customer authentication; *a circle does not mean that a type of information was necessary or by itself sufficient for SIM swap customer authentication.*

Although within each carrier the set of authentication mechanisms used by the 10 CSRs were mostly consistent, there was no particular pattern in which they were presented to us. The one exception, however, was T-Mobile: the order of PIN, OTP, and call log was consistent through all 10 calls. Further, providers that support PIN authentication (AT&T, T-Mobile, Tracfone, and Verizon) always used that mechanism first.

Our key findings are as follows:

1. **Mobile carriers use insecure methods for authenticating SIM swaps.**

   a. **Last payment.** We found that authenticating customers via recent payment information is easily exploitable. AT&T, T-Mobile, Tracfone, and Verizon use payment systems that do not require authentication when using a refill card. An attacker could purchase a refill card at a retail store, submit a refill on the victim's account, then request a SIM swap using the known refill as authentication.

   b. **Recent numbers.** We also found that using information about recent calls for authentication is exploitable. Typically CSRs requested information about *outgoing* calls. Consider the hypothetical following attack scenario:

Using only the victim's name and phone number, our simulated adversary could call the victim and leave a missed call or message that would prompt the victim into returning the call to a number known to the attacker. This call would then appear on the outgoing call log and the attacker could use it for authentication. CSRs appeared to also have the discretion to allow authentication with *incoming* call information, as this occurred four times between AT&T, T-Mobile, and Verizon. An attacker can trivially generate incoming call records by calling the victim.

c. **Personal information.** We found that Tracfone and US Mobile allowed personal information to be used for authentication. While our simulated attacker did not use this information, it would likely be readily available to real attackers (e.g., via data aggregators) and is often public, so it offers little guarantee of the caller's identity. We note that for over a decade, FCC rules have prohibited using "readily available biographical information" to authenticate a customer requesting "call detail information."[3]

d. **Account information.** We found that AT&T, US Mobile, and Verizon allowed authentication using account information. As with personal information, this information would often be readily available to an adversary. Receipts (whether physical or electronic), for example, routinely include the last four digits of a payment card number. We note that PCI DSS, the industry standard for protecting payment card information, does not designate the last four digits of a payment card as "cardholder data" or "sensitive authentication data" subject to security requirements [53]. As for the activation date associated with an account, that information may be readily available from business records (e.g., via a data aggrega-

[3] 47 C.F.R. § 64.2010.

tor), inferable by website or mobile app logs (e.g., via User-Agent logs), or inferable via mobile app API access (e.g., via the usage stats API on Android or the health APIs on Android and iOS). We note that FCC rules also prohibit using "account information" to authenticate a customer requesting "call detail information."[4]

   e. **Device information.** We found that all carriers except for T-Mobile use device information for authentication. These authentication methods included the customer's IMEI (device serial number) and ICCID (SIM serial number). Both the IMEI and ICCID are available to malicious Android apps, and IMEIs are also available to adversaries with radio equipment.

   f. **Security questions.** We found that Tracfone used security questions for authentication. We also found that T-Mobile, Tracfone, and Verizon prompted users to set security questions upon signup. Prior research has demonstrated that security questions are an insecure means of authentication, because answers that are memorable are also frequently guessable by an attacker [35, 54, 55].

2. **Some carriers allow SIM swaps without authentication.** Tracfone and US Mobile did not offer any challenges that our simulated attacker could answer correctly. Yet, CSRs at these carriers allowed us to SIM swap without ever correctly authenticating: six times at Tracfone and three times at US Mobile.

3. **Some carriers disclose personal information without authentication, including answers to authentication challenges.**

   - **AT&T.** In one instance, the representative disclosed the month of the activation and last payment date and allowed multiple tries at guessing

[4] *Id.*

the day. They also guided us in our guess by indicating whether we were getting closer or further from the correct date.

- **Tracfone.** In one instance, the representative disclosed the service activation and expiration dates. Neither are used for customer authentication at Tracfone.
- **US Mobile.** In three instances, the representative disclosed the billing address on the account prior to authentication. In one instance, a portion of the address was leaked. In one instance, part of the email address was disclosed. In three instances, the representative disclosed portions of both the billing address and email address.

Table 3.2.: The outcomes of our SIM swap requests. Note that our attempts at major carriers were all successful.

	AT&T	T-Mobile	Tracfone	US Mobile	Verizon
Success	10	10	6	3	10
Failure	0	0	4	7	0

Table 3.3.: The authentication scheme that was used to authenticate the calls on successful attempts.

	Recently dialed numbers	Last payment details	No authentication
AT&T	2	8	0
T-Mobile	10	0	0
Tracfone	0	0	6
US Mobile	0	0	3
Verizon	9	1	0

In addition to learning the carriers' authentication policies, we also documented whether the swap was successful or not. The outcomes are shown in Table 3.2.

In our successful SIM swaps, we were able to authenticate ourselves with the carrier by passing at most one authentication scheme. For instance, Verizon—a provider that uses call log verification—allowed us to SIM swap once we provided two recently dialed numbers, despite us failing all previous challenges, such as the PIN. Some CSRs at Tracfone and US Mobile also forgot to authenticate us during our calls, but they were able to proceed with the SIM swap, indicating that back-end systems do not enforce authentication requirements before a customer's account can be changed. Table 3.3 details the exact authentication challenge that was exploited in each successful call.

Devices transmit identifying information to the network, namely the International Mobile Equipment Identity (IMEI), which is unique to the device. Therefore carriers could presumably detect that we were not only switching SIM cards, but devices as well. This never presented an issue across our 50 calls; in three cases, the CSR noted verbally that the device IMEI had changed, but did not intervene or flag the account.

Our key finding is that all three major carriers in our study used manipulable information—call logs and/or payment information—for authentication. Carriers may have changed their customer authentication practices since our testing. We requested that they update us if they did.

3.6. Discussion

3.6.1. Weak authentication mechanisms

It has long been known that carriers' authentication protocols are subject to social engineering or subversion using stolen personal information [56, 57]. We found an additional, more severe vulnerability: carriers allow customers to authenticate using information that can be manipulated without authenticating.

In our experiments, several carriers relied on call log verification as an authentication method, asking us to provide recently dialed phone numbers (T-Mobile asked only for the last four digits of one recently dialed number; Verizon required two full phone numbers). An adversary could easily obtain these records by baiting victims into calling numbers that he knows about. As an example, the adversary could first send an intentionally vague text message claiming to be an institution that the victim frequents (e.g., her school, bank, or healthcare provider) with a callback number. The victim might then call the number to learn more details. As long as the call connects, an outgoing call to this number will be logged in the victim's call record. The adversary can then provide that number as a correct response to the challenge when requesting a SIM swap at the carrier. Another attack that achieve the same result is the "one-ring" scam, in which the attacker hangs up just as the victim's phone starts ringing; the victim—upon seeing the missed call—will call back out of curiosity. To make matters worse, in four instances between AT&T, T-Mobile, and Verizon, we were able to succeed call record verification by providing incoming numbers. This means that the adversary would not even need the victim to place a call; as long as the victim picks up the initial call from the adversary, a valid record in the call log would be generated.

The second manipulable authentication challenge we saw in our experiments is payment record verification. In these cases, we were asked to provide details about the most recent payment on the accounts. Most of the carriers in our study—including all of the major carriers—allow for payments to be made over the phone. None of these payment systems require any authentication when making these payments using a refill card, even when calling from a third-party number. To obtain payment information, an adversary can first purchase a refill card for the victim's mobile carrier at, for example, a convenience store. After dialing into the payment system, he can enter the victim's phone number and redemption code on the refill card to add value to her account. Once the payment is accepted, the adversary—now with complete knowledge of the most recent payment—can call the carrier to request a SIM swap and successfully pass payment record verification. This attack has an even lower barrier to entry than call log verification because it requires no action from the victim. Although it does require the attacker to spend a small amount of money, minimum required payments are typically quite low (between $5-30 in our experiments). As shown in Table 3.1, two of the five carriers in our study (both major carriers) support payment record verification. For AT&T, payment record verification was used consistently in all 10 calls. Only US Mobile did not allow for unauthenticated refills to be made; they only supported online refills which required account authentication.

Tracfone and US Mobile—the MVNOs—did not use any manipulable information for authentication and thus had fewer successful swaps. However, nearly all of their authentication challenges came from public records. A dedicated adversary would plausibly be able to obtain a victim's DOB, address, email address, or answers to security questions through online profiles, and thus be able to successfully authenticate at the carriers. Even then, we were still able to succeed at Tracfone and US Mobile in instances where CSRs skipped authentication, which suggests that policies for customer authentication at those carriers might not be as rigorous as those at other carriers.

In all instances of unauthenticated information leakage, the customer service representatives had released parts of the answer—either the email address, billing address, activation date, or payment date—as hints and said we would be authenticated once we remembered the whole response. This suggests that sensitive account details are stored in the clear and visible to CSRs, who are thus susceptible to social engineering attacks.

3.6.2. Severity

It has long been known that mobile subscribers are at risk of SIM swap attacks [58–60]. Our research demonstrates that insecure means of customer authentication are still widely used by mobile carriers. This exposes customers to severe risks: denial of service, interception of sensitive communications, and impersonation, which can lead to further account compromises.

As mentioned above, an attacker who hijacks a victim's phone number could intercept authentication passcodes sent by SMS or phone call. Phone-based passcode authentication as a second factor or account recovery method is ubiquitous on the internet, including at financial institutions and cryptocurrency exchanges where access to online accounts confers access to funds. Since reports about bank theft stemming from SIM swap attacks appear regularly in the media, we consider this a high severity vulnerability [12, 61].

At the recommendation of wireless carriers, we conducted an additional round of data collection to understand how customers could protect themselves against SIM swap attacks. We signed up for one additional prepaid account each with AT&T, T-Mobile, and Verizon; after one week, we called to inquire about and enable any safeguards against SIM swaps and port outs, citing T-Mobile's `NOPORT` as an example.[5] None of the carriers had additional protection features beyond the ones we had set in our initial study. We placed these calls in September 2019.[6] This additional result indicated that prepaid customers not only were vulnerable to SIM swap attacks, but also were not capable of easily employing any mitigation.

[5] `NOPORT` is a T-Mobile option that heightens authentication requirements for port out requests [62]. While `NOPORT` would not itself protect against SIM swap attacks, at least as currently implemented, we referenced it during our calls with CSRs. During the course of our additional data collection, we also found that T-Mobile did not offer `NOPORT` for prepaid accounts.

[6] Verizon has since implemented an opt-in feature called `Number Lock`, which prohibits port out requests unless switched off [63]. The feature is exclusive to postpaid accounts.

We studied prepaid accounts because they can be registered without undergoing a credit check, enabling us to scale the number of test accounts. Prepaid plans accounted for 21% of U.S. wireless connections in Q3 2019, or about 77 million connections [64].[7] Compared to postpaid accounts, these contract-free plans are less expensive and do not require good credit, so they are more attractive to (and are often marketed to) low-income customers. Based on our experimental results for prepaid accounts, as well as our anecdotal evaluation of postpaid accounts (presented in § A.1), we hypothesize that current customer authentication practices disproportionately place low-income Americans at risk of SIM swap attacks.

Anecdotally, during this study, one of the authors themselves fell victim to an account hijacking via a SIM swap attack. After initial unsuccessful attempts to authenticate himself to the carrier using personal and knowledge-based information, he escalated the issue to the carrier security team. From there, he was able to leverage our findings by requesting to authenticate via recently dialed numbers—a method which we knew the carrier supported although it had not been offered in this instance.

3.7. Analysis of phone-based authentication

Software tokens and SMS-based passcodes delivered by SMS or call have become popular authentication schemes for online services [67, 68]. SMS-based passcodes as a second authentication factor are an especially common option, as they make the security of MFA available to any user with an SMS-enabled phone.

[7] This figure is based on data from carriers' earnings and financial statements. Carriers may use slightly different terms and definitions; e.g., Verizon defines a "connection" as an individual line of service for a wireless device while T-Mobile defines a "customer" as a SIM card associated with a revenue-generating account [65, 66], a seemingly equivalent metric. These definitions explain how carriers appear to have a population penetration rate above 100%, as an individual can possess multiple wireless-connected devices.

We aimed to reverse-engineer the authentication policies of popular websites and determine how easy it is for an attacker to compromise a user's account on the website provided they have successfully carried out a SIM swap.

3.7.1. Method

We started with the dataset used by `TwoFactorAuth.org`, an open-source project to build a comprehensive list of sites that support MFA. Anyone can contribute MFA information about websites to the database, while the owner—a private developer—acts as the moderator. In the dataset, over 1,300 websites are grouped by categories including healthcare, banking, and social media. The available methods are also listed under each website in the dataset. As of late 2019, 774 of the sites in the dataset support MFA; of those, 361 support SMS-based MFA. The 361 websites that support SMS-based authentication are of interest to us. Of these, 145 were accessible for our analysis; the rest required ID verification, enterprise signups, payment, or were duplicate entries (e.g., the Xbox site uses Microsoft's login system). We used a snapshot of the dataset from November 1, 2019.

The `TwoFactorAuth.org` dataset lists the available authentication factors for each website, but it does not include information about how authentication can be configured or how different authentication factors are presented to the user (e.g., which are recommended or set as defaults). To compile this information, we signed up for accounts at each website and traversed their authentication flows. To the best of our knowledge, we contribute the first dataset that shows how MFA is implemented in practice.

At each website, we created a user account and provided all requested personal information. After signing up, we enrolled in MFA using the recommended configurations at each site, opting for schemes that were mandated, listed first, or had conspicuous labeling. We then examined other possible MFA configurations, if available, taking note of schemes that were mandatory, linked, or automatically activated. Between each configuration setup, we also looked at account recovery options. We took screenshots of the authentication options, enrollment process, login procedures, and account recovery procedures at all websites. We tested each configuration on a new browser session with no previous site data.

We classified configurations into three categories: secure, insecure, and doubly insecure. A doubly insecure configuration indicates that a SIM swap alone is enough for account compromise; the configuration uses both SMS-based MFA and SMS-based password recovery. An insecure configuration can only be compromised if the attacker knows the account password; these configurations offer SMS-based authentication but do not allow for SMS-based password recovery (the attacker could obtain the password via data dumps, social engineering, or compromising the victim's account recovery email). The secure configuration uses stronger authentication schemes, such as authenticator apps, and cannot be recovered or reset by SMS.

3.7.2. Results

Our key findings are as follows:

1. **The majority of websites default to insecure configurations.** Of the 145 websites, 83 (a majority) have recommended or mandated configurations that are insecure. For most of these websites, there are other secure schemes present; only 14 websites have SMS as their sole MFA option.

2. **Some websites are doubly insecure.** 17 websites allow doubly insecure configurations, 13 of which default to or recommend doubly insecure configurations.[8] Accounts of users who choose these configurations can be compromised with a SIM swap alone. That is, an attacker needs only the victim's phone number to reset the password and bypass SMS-based authentication. These websites span different industries, including finance (Paypal, Venmo, Taxact), travel (Finnair), commerce (Amazon, eBay), and social media (Snapchat). We initially redacted the names and other identifying information of these websites in our annotated dataset, while providing initial notification as part of the responsible disclosure process (§ 3.7.3).

   Recall that the doubly insecure configuration is only possible if SMS-based account recovery is also available. We found 11 websites that use SMS-based password recovery, but switch to different recovery tools—such as email or manual review—when MFA is enabled. Similarly, we found two websites that switch when SMS-based MFA is enabled.

3. **Security is only as good as the weakest link.** 10 websites recommend secure authentication schemes but simultaneously suggest insecure methods, like SMS or personal knowledge questions, as backups. Since an attacker only needs to defeat one of the authentication schemes to defeat MFA, an insecure backup renders the configuration insecure. Eight websites with multiple authentication options also mandate initial enrollment in SMS before allowing users to switch to other MFA schemes. Six websites with multiple options mandate SMS in order to keep MFA enabled.

[8] Additionally, 10 websites that have SMS-based password recovery from examining their account recovery pages, but could not sign up for accounts due to the aforementioned restrictions.

4. **Some websites give users a false sense of security.** Some services automatically enroll users in email- or SMS-based MFA using the email address or phone number on file, respectively, without any user input or notice. Seven websites enroll users in SMS-based MFA without notice, either with the account recovery number or a phone number a user must provide in order to sign up for a non-SMS-based 2FA method. Even if the user then signs up for another MFA method, they continue to be simultaneously enrolled in SMS-based MFA without being made aware of it. Thus even users who are educated about SIM swap risks may nonetheless be lulled into a false sense of security. At four of these websites, the automatic SMS 2FA enrollment renders the configuration doubly insecure. A user may believe that account compromise requires both a stolen password and a compromise of the authenticator app (e.g. via phone theft), but in fact, a SIM swap alone is sufficient.

5. **Some websites offer 1-step SMS OTP logins.** Seven websites also offer 1-step logins via an SMS OTP. eBay, for instance, will send users a temporary password via SMS if MFA is not enabled, and WhatsApp uses SMS OTP by default if MFA is not enabled.

The annotated dataset describing all of our findings is available at `issms2fasecure.com`.

3.7.3. Failures in vulnerability disclosure processes

We attempted to responsibly disclose the vulnerabilities we uncovered to the 17 affected websites. Only in 4 of the 17 cases did the process work as expected and result in bug fixes. We document the failures we encountered and call for improvements in vulnerability disclosure processes.

Method. In January 2020 we attempted to notify the 17 websites described above of the presence of doubly insecure configurations. We first looked for email addresses dedicated to vulnerability reporting; if none existed, we looked for the companies on bug bounty platforms such as HackerOne. Many companies outsource bug reporting to these third-party platforms in order to triage reports for relevance and novelty. Reports are screened by employees of the platform, who are independent from the company, and passed on to the company's security teams if determined to be in scope. If we were unable to reach a company through a dedicated security email or through bug bounty programs, as a last resort, we reached out through customer support channels.

Sixty days after our initial notifications, we re-tested the companies using the same method in § 3.7.1, except for those that reported that they had fixed the vulnerabilities.

Outcomes. Three companies—Adobe, Snapchat, and eBay—acknowledged and promptly fixed the vulnerabilities we reported. In one additional case, the vulnerability was fixed, but only after we exhausted the three contact options listed above and reached out to company personnel via a direct message on Twitter.

In three cases—Blizzard, Microsoft, and Taxact—our vulnerability report did not produce the intended effect (as documented in the following paragraph), but in our 60-day re-test we found that the vulnerabilities had silently been fixed. We do not know whether the fixes were implemented in light of our research.

There were several failure modes, which were not mutually exclusive.[9] In five cases, personnel did not understand our vulnerability report, despite our attempts to make it as clear as possible, shown in § A.3. For example, Microsoft claimed that SIM swaps are widely known, and did not appreciate that their insecure MFA configuration exacerbated the issue. In five cases, we received no response. Predictably, all four attempts to report security vulnerabilities through customer

[9] The counts in this paragraph are out of a total of 13 websites, including the three that silently fixed the vulnerabilities.

support channels were fruitless: either we received no response or personnel did not understand the issue. Three of the four reports we submitted to bug bounty programs also resulted in failures and were closed due to the absence of a bug (recall that our findings are not software errors, but rather, logically inconsistent customer authentication policies).[10] HackerOne employs mechanisms that restrict users from submitting future reports after too many closed reports [69], which could disincentivize users from reporting legitimate vulnerabilities [70].

We have listed all 17 responses in § A.3. Unfortunately, nine of these websites are doubly insecure *by default* and remain so as of this writing. Among them are payment services PayPal and Venmo. The vulnerable websites cumulatively have billions of users.

We provide an up-to-date timeline of responses on this study's website at `issms2fasecure.com`.

3.7.4. Analysis of enterprise MFA solutions

Many organizations offer (or require) MFA to their personnel for accessing internal resources. Most of these MFA solutions are provided by third-party services and integrate with organizations' existing login pages. To further understand the downstream impact of SIM swaps, we examined the handling of SMS-based MFA by three such vendors: Duo Security, Okta, and Microsoft. We selected these solutions based on popularity reports by Gartner, a global technology research and advisory firm [71]. We focused on the security-usability tradeoff provided by these solutions.

Method. In addition to checking the documentation for how those services handle SMS-based MFA, we created fictitious organizations and signed up for administrator

[10] We had unsuccessfully submitted our vulnerability reports to carriers via HackerOne when possible (i.e. for AT&T, T-Mobile, and Verizon) before reaching out to CTIA. If we include those figures, six out of seven reports to bug bounty programs resulted in failures.

accounts at each service. Next, we invited a new user to our organization, and finished account setup—along with MFA enrollment—from the user view. Both services offer proprietary mobile apps that come with authentication prompts and authenticator passcodes (TOTP); we installed the apps when instructed. Our findings are as follows:

Findings: Duo Security MFA. We find that Duo automatically and silently enrolls the user in SMS-based MFA, despite the availability of stronger second factors, unnecessarily weakening security.

When a user enrolls in MFA, Duo requires them to specify the type of device they are adding. If the user elects to add a smartphone (which Duo recommends), she will be required to add a phone number.[11] The user will be automatically enrolled in SMS-based MFA, provided that the organization has enabled it (which is the default). The user is also automatically enrolled in two other MFA methods: push notifications and TOTP. Users are not informed of the authentication methods they have been enrolled in during setup.

Users can view their authentication methods after logging in for the first time by navigating to the MFA page. However, they cannot modify their authentication methods (e.g., disable SMS-based MFA) — only an administrator can do so. Intriguingly, we found that users can bypass the requirement to enter a phone number (while retaining the other authentication methods) by setting up their smartphones as tablets. However, this is undocumented.

Findings: Okta Adaptive MFA. Okta does not suffer from the abovementioned vulnerability. It uses a method-oriented enrollment process: users explicitly enroll in authentication methods without being asked to provide their device details.

[11] Duo allows administrators to add devices for users in the admin interface, where the phone number requirement for smartphones is also present.

However, only the proprietary app is enabled as a second factor by default, while all other authentication methods, including SMS, are disabled, which means that users are not given any choice of authentication methods and cannot choose to enroll in SMS-based 2FA. Unless an administrator changes this policy, users without smartphones — or who do not wish to install the app — are locked out of the system.

Findings: Microsoft Azure MFA. We find that Azure defaults to SMS-based MFA during enrollment, despite the availability of stronger second factors, potentially weakening security.

Azure—like Okta—uses a method-oriented enrollment process. With the default administrator settings, users are able to choose between SMS, push notifications, and TOTP, with SMS being the default. However, the UI is slightly confusing: users must first select the medium to receive authentication messages from a dropdown menu (e.g., "Authentication phone" for SMS, "Mobile app" for push notifications and TOTP). "Authentication phone" is the default menu option provided that the organization has enabled SMS-based MFA (which is the default), so a user may be unaware that stronger second factors are available.

The contrasting approaches by Duo and Okta, and their corresponding limitations — one weakens security, and the other hurts usability — suggests an underlying issue, which is that the MFA vendors seek to maximize administrators' control over configuration for the whole organization and minimize variation between users. Allowing users more control, while also giving them guidance about benefits and risks, may allow for a more nuanced security-usability tradeoff. Azure does give users such control, although it offers SMS as the default and the confusing user interface compounds this issue.

3.8. Recommendations

3.8.1. Recommendations for carriers

In evaluating existing and proposed authentication schemes, we looked to the framework proposed by Bonneau et al. to consider the usability, deployability, and security of these mechanisms [72]. We also discussed usability and deployability issues with wireless carriers and CTIA. We offer the following recommendations:

1. **Carriers should discontinue insecure methods of customer authentication.** Every mobile carrier in our study, with one exception, already offers secure methods of customer authentication: password/PIN,[12] one-time passcode via SMS (to the account phone number or a pre-registered backup number), or one-time passcode via email (to the email address associated with the account). Abandoning insecure authentication schemes—personal information, account information, device information, usage information, and security questions—may inconvenience customers who are legitimately requesting a SIM swap, but preventing account hijacking attacks is crucial to customers' privacy and security. Moreover, legitimate SIM swap requests appear to be infrequent, occurring only when a user's SIM is damaged or lost, when a user acquires a new phone that is incompatible with their SIM, or in other rare cases. These requests may become even more infrequent going forward, as users are now waiting longer before switching their devices [74]. Thus, carriers should begin to phase out insecure authentication methods and develop measures to educate customers about these changes to reduce transition friction. Carriers should use data on the type and frequency of legitimate SIM swaps to assess the usability impact of authentication procedures.

[12] A password or PIN that is easily guessed is not secure, of course. Carriers must have safeguards that prevent users from choosing weak PINs [73].

2. **Implement additional methods of secure customer authentication.** We recommend that mobile carriers implement customer authentication for telephone support via a website or app login, or with a one-time password via a voice call. The methods do not require memorization or carrying extra devices and are easy to learn. They also should not pose significant costs to carriers because the infrastructure already exists; all carriers we examined support online accounts via websites and/or mobile applications.

3. **Provide optional heightened security for customers.** We recommend that carriers provide the option for customers to enable MFA for account change requests, as well as the option to disable account changes by telephone or at a store.

4. **Respond to failed authentication attempts.** If someone attempts to authenticate as a customer and is unsuccessful, we recommend that carriers notify the customer and heighten security for the account. An adversary should not be allowed to attempt multiple authentication methods or to repeatedly attempt authentication. Moreover, even if an adversary was able to successfully authenticate after failing previous attempts, carriers should not be convinced that the caller is who they claim to be. For instance, a customer who has forgotten their PIN, is unable to access their email and backup phone for an OTP, but can recall some call log information, is very unlikely to be the customer, but rather an adversary who is trying to authenticate using call log verification. If a customer who loses or has their phone stolen goes into a store and attempts to purchase a new device with the original number, they should not be allowed to authenticate with only a government-issued ID. IDs are open to forgery, and the absence of the original device—though unfortunate—should result in additional security measures being taken. In both scenarios, the carrier

can respond in different ways, such as adding a 24 hour delay to a SIM swap request while notifying the customer via SMS or email, going further down the authentication flow, or denying the caller's request for a period of time. In other words, authentication should not be binary.

5. **Restrict customer support representative access to information before the customer has authenticated.** There is no need for representatives to access customer information before authentication, and providing such access invites deviation from authentication procedures and enables social engineering attacks. In all instances of unauthenticated information leakage in our study, the customer support representatives had released parts of the answer as hints and stated we would be authenticated once we remembered the whole response. This strongly suggests that sensitive account details are, for at least some carriers, visible to representatives prior to customer authentication.

6. **Publicly document customer authentication procedures.** Carriers should list all the ways customers can be authenticated over the phone in order to avoid uncertainties regarding risks and defenses. They also stand to benefit from informing their customers and homogenizing the authentication flow within and between carriers. In addition, carriers should maintain pages that explain SIM swap attacks and any available security countermeasures that they offer.

7. **Provide better training to customer support representatives.** Representatives should thoroughly understand how to authenticate customers and that deviations from authentication methods or disclosure of customer information prior to authentication is impermissible. That said, we emphasize that training alone is not sufficient—there should also be technical safeguards in place.

Taken collectively, these recommendations should decrease the number of unauthorized SIM swaps by improving user authentication.

3.8.2. Call for research: better design of customer service interfaces

It is essential that authentication procedures be consistent across callers and CSRs. This is challenging because CSRs may be susceptible to social engineering attacks (e.g., an adversary pretending to be a victim of domestic violence desperate to urgently regain control over their account). The software used by CSRs play an integral role in keeping accounts secure, in particular:

1. Restrict CSR access to account information before the customer has authenticated
2. Present authentication mechanisms in a consistent order
3. Prohibit CSR bypass of user authentication

From our study, we believe that current customer support interfaces do not meet the above-mentioned requirements. That is, CSRs released parts of the answer as hints, authentication mechanisms were generally not presented in any particular order within and across carriers (with the exception of T-Mobile), and carriers allowed us to SIM swap without ever correctly authenticating in nine instances (§ 3.5). We are unable to find information about any software tools used by CSRs for authenticating customers.

An improved secure interface should complement improved CSR training procedures, and more importantly, be easy for CSRs to use. To our knowledge, CSRs themselves have never been subjects of study from a security and usability perspective. Just as the security community has realized the value of research on developers making security design decisions, CSRs should also be subjects of research, in order to effectively study security in practice [75, 76]. By studying workers' behaviors, the community can make recommendations on training procedures and interface design.

Our suggestions above can only be implemented with commitment from the carriers themselves. We call on carriers to collaborate with usable security researchers to study CSRs and their software tools. One important open research question is how carriers should respond to failed authentication attempts. Ignoring failures carries security risks (as we have documented) but an overly strict policy risks locking out customers. In the long term, carriers (and all other organizations that need to authenticate customers over the phone) should endeavor to develop an industry standard, informed by research, that is accessible to the community for scrutiny.

3.8.3. Recommendations for websites

Carriers are ultimately responsible for mitigating the authentication vulnerabilities that we have reported, but meanwhile, users of websites relying on SMS-based MFA continue to be at risk—in some cases severely (§ 3.7.2). We offer the following recommendations for websites to better protect their users from the effects of SIM swap attacks:

1. **Employ threat modeling to identify vulnerabilities.** Threat modeling is a fundamental information security technique that is used to identify vulnerabilities in a systematic way. It consists of a structured analysis of the application, the attacker, and the possible interactions between them. Many of our findings, especially the existence of doubly insecure websites, suggest a failure (or absence) of threat modeling.

2. **Implement at least one secure MFA option.** Websites without any other MFA options should roll out alternative options such as authenticator apps, and notify users when these options become available. Popular secure MFA options do not pose large usability hurdles. Reese et al. performed a usability

lab study of five 2FA methods, including push notifications, SMS, TOTP, and U2F [77]. They found—with statistical significance—that push notifications, TOTP, and U2F have faster median authentication times and higher system usability scale (SUS) scores than those of SMS. Authenticator apps also have an added usability benefit over SMS-based MFA: the device need not be online to generate the one-time password.

3. **Eliminate or discourage SMS-based MFA.** Websites should not make SMS the default or recommended MFA option. Websites should highlight the dangers of SIM swaps, and label SMS as an option with known risks. As of 2019, only 15% of adults in the U.S. own non-smartphone cellular devices (compared to 81% of adults in the U.S. that own smartphones) [78]. As that share continues to decrease, websites should eliminate SMS-based MFA altogether.

4. **Improve vulnerability disclosure processes.** A bug bounty program is not a substitute for a robust security reporting mechanism, yet some companies are using it as such (Section 3.7.3). These third-party platforms appear to be overly strict with their triage criteria, preventing qualified researchers from communicating with the companies. Companies should maintain direct contact methods for security reporting procedures.

3.9. Summary

The theory and practice of user authentication has come a long way in the last decade. Yet these gains have been uneven. We found that five carriers in the United States continue to use authentication methods that are now known to be insecure, enabling straightforward SIM swap attacks. Further difficulties arise when security rests on interactions between independent systems. Phone-based authentication, and SMS in particular, has made rapid inroads because of convenience, but carriers don't adequately account for this scope creep in protecting against SIM swaps. Meanwhile, many online services view SIM swaps as "someone else's problem."

In addition to fixing the vulnerabilities we identified, our work suggests fruitful avenues for academia and industry: better quantifying the security-usability tradeoff in specific settings including over-the-phone authentication and enterprise authentication; studying user populations such as customer-service representatives and their user interfaces; and improving the vulnerability disclosure process for non-software vulnerabilities.

4

Security and Privacy Risks of Number Recycling at Mobile Carriers in the United States

4.1. Introduction

Recycled phone numbers can cause trouble for all those involved. Subscribers who are assigned a previously owned phone number often end up receiving communication meant for the previous owners, from threatening robocalls to personal text messages. One journalist, right after changing her number, was bombarded with texts containing blood test results and spa appointment reservations, while another accidentally wound up in a previous owner's email inbox after requesting a login passcode via SMS [79, 80]. A recent survey of 195 participants found these incidents are common; 72 reported negative experiences related to number recycling, including dealing with communication meant for previous owners [81]. While neither the journalists nor any of the study participants had any malicious intent, this naturally raises concerns about adversaries exploiting these incidents for gain.

In this study, we present eight different attacks enabled by phone number recycling. Of those, we empirically evaluated three low-cost attacks that allow new owners of recycled numbers to compromise the security and privacy of previous owners. We analyzed the set of phone numbers available through the online interfaces of two U.S. mobile carriers: T-Mobile and Verizon Wireless. By analyzing the structure of phone number blocks that contain primarily recycled versus primarily fresh numbers, we developed a strategy for the adversary to focus their attention on the former. Our key finding is that most of the available phone numbers we sampled (215 of 259) were recycled and also vulnerable to one or more of the three number recycling attacks.

Throughout our study, the adversary only needs to interact with standard online number change interfaces to carry out these attacks, and does not need to exploit software vulnerabilities. We found that the online interfaces in question imposed few restrictions on the adversary's ability to browse and obtain previously owned numbers for exploitation. We estimate the number of available recycled phone numbers at Verizon to be about one million, with a largely fresh set of numbers becoming available every month.

We found that carriers did not proactively notify subscribers about their policies regarding number recycling. Worse, they provided inconsistent responses when asked. We called in to customer service to ask about number aging periods—the time before a disconnected number is made available again. We received widely divergent answers at each carrier (seven unique responses out of 13 calls to T-Mobile, eight unique responses out of 13 calls to Verizon). Subscriber confusion or unawareness of recycling policies could be one reason why the vulnerabilities we document are so prevalent.

Finally, we obtained and monitored 200 recycled numbers from both carriers. With just one week of data, we conservatively found nearly 10% of numbers in our honeypot were still receiving security/privacy-sensitive communications meant for previous owners. Upon receiving these unsolicited calls / texts, owners of recycled numbers can suddenly realize the incentives to exploit and become opportunistic adversaries. Due to our limited monitoring period, the actual proportion of vulnerable numbers is likely much higher.

As the number of users coming online continues to grow, number recycling threats are unlikely to abate. Phone numbers have become tied to peoples' identities more than ever, through social media accounts, ridesharing apps, mobile banking, etc. They are used to link online accounts to real-world entities and for authentication. Unfortunately, numbers are a finite resource. In the United States, when a subscriber gives up their 10-digit phone number, it eventually gets reassigned to someone else. While carriers, websites, and subscribers can take steps to reduce risk, number recycling threats highlight fundamental problems with the use of phone numbers for security-sensitive purposes.

Responsible disclosure and responses. In October 2020 we provided an initial notification of our findings to the carriers we studied and to CTIA, the U.S. trade association representing the wireless communications industry.

In December 2020, T-Mobile informed us that after reviewing our research, it had updated its number change support page to 1) remind subscribers to update their contact number on bank accounts and social media profiles, and 2) specify the FCC-mandated number aging period. Along with raising subscriber awareness, it also informed us that customer service agent manuals had been updated to emphasize those two points during relevant interactions, effective early December.[1]

In December 2020, CTIA informed us that after reviewing our research, Verizon had updated its public-facing support document for number cancellations, suspensions, and transfers to 1) remind subscribers to update their contacts and unlink

[1] `https://www.t-mobile.com/support/account/change-your-phone-number`. Visited 03/22/2021.

their business and online accounts, and 2) specify the FCC-mandated minimum aging period (45 days).[2]

Social impact. In March 2021, we reached out to academic researchers studying technology-enabled intimate partner violence (IPV), and discussed the harms of number recycling attacks targeting survivors of IPV.[3] The team is currently drafting an update to their clinic resources to include our research and recommendations.[4]

4.2. Background

4.2.1. Phone-based authentication is prevalent

According to 2FA Directory—a crowd-sourced project to build a comprehensive list of sites that do or do not support multi-/two-factor authentication (2FA), about 30% of websites (455 / 1,565) support SMS-based authentication as of January 2021 [82].[5] Its popularity is only surpassed by that of authenticator apps, which is present at 40% (626/1,565) of websites. 957 websites in the dataset support at least one form of 2FA.

By SMS-based authentication, we mean the method of sending a single-use passcode (OTP) to the subscriber's phone via an SMS text message or a phone call. This type of authentication is vulnerable to phone line changes because they are tied to a phone number and the associated cellular service. Other types of phone-based authentication (e.g., authenticator apps) are not vulnerable to phone line changes.

Phone numbers themselves are regularly used by systems to authenticate callers. Some automated customer service phone systems—such as for credit cards—

[2] `https://www.verizon.com/support/cancel-suspend-transfer-lines/#change`. Visited 03/22/2021.

[3] `https://www.ipvtechresearch.org/`

[4] `https://www.ceta.tech.cornell.edu/resources`

[5] Anyone can contribute 2FA information about websites, while a group of private developers acts as the moderator. As such, the 1,565 websites should be viewed as a convenience sample.

automatically announce sensitive account information if the caller ID corresponds to an existing profile, without any subscriber input [80]. Even after a phone line change, these external systems can continue to reveal a previous owner's credit card or utility account information to the new owner of the phone number, unless the previous owner manually updates their contact.

4.2.2. Subscribers may give up or lose their phone number for many reasons

According to the Federal Communications Commission (FCC), around 35 million phone numbers in the U.S. are disconnected every year [41]. At the end of 2018 (the latest published data at time of writing), there were more than 860 million phone numbers in use by active subscribers [83].

People may give up their phone number for various purposes, such as to:

1. Prevent unwanted parties from contacting them (e.g., abusive acquaintances, collections agencies)[6]
2. Switch to a new carrier[7]
3. Cancel telephone service altogether (e.g., moving out of the country, switching to a job-provided phone account)
4. Switch to a more desirable number [84]

Subscribers may also lose their account and their phone number due to:

1. Nonpayment

[6] https://www.reddit.com/r/legaladvice/comments/bs2nbv/help_me_take_legal_action_against_my_ex_who_has/. ("A month after a nasty breakup, I told my abusive ex to never contact me again... I have to change my phone number because of him.")

[7] Most carriers are required to allow active departing subscribers to bring their numbers to their new carriers. Of course, subscribers may elect to receive new numbers, thereby releasing their original ones.

2. Violation of service terms

3. Inactivity (e.g., Google Voice [85], Twilio [86])

4.2.3. Most relinquished phone numbers get reassigned

Most relinquished numbers are not permanently retired. There is only a finite number of 10-digit phone numbers; all will eventually be assigned to carriers, thereby capping expansion. Since the FCC assigns phone numbers to carriers in contiguous blocks of 1,000 rather than individually, it has sought to forestall exhaustion for as long as possible by activating fresh blocks of phone numbers only when absolutely necessary.[8] To that end, it has enacted policies to prevent carriers from hoarding numbers and encourage carriers to routinely recycle numbers by assigning them to new subscribers after a waiting period [87]. As a result, new owners of previously-assigned numbers often end up receiving personal communication meant for the previous owners.

4.2.4. Number recycling is regulated by the FCC

There are also FCC rules specific to number recycling that aim to encourage carriers to recycle numbers while mitigating the risks to subscribers. However, the only risk that the FCC appears to be concerned about is that of receiving robocalls meant for previous owners, and not any of the other threats we discuss here.

Carriers are prohibited from reassigning disconnected numbers until 45 days have elapsed since disconnection, and can age numbers for up to 90 days (365 days for numbers assigned to business customers).[9]

[8] One of the reasons to prolong the usefulness of 10-digit dialing is the exorbitant cost of adding another digit; many existing automated devices are only programmed to handle 10-digit phone numbers.

[9] 47 C.F.R. § 52.15(f)(1)(ii)

In December 2018—in efforts to combat unlawful robocalls—the FCC announced a plan to create a reassigned number database (RND), along with establishing the 45-day minimum aging period [42]. Carriers would be mandated to report recycled numbers on a monthly basis, which would be compiled into a centralized source for legitimate robocallers (e.g., refill prescription reminders) to reference. Carriers were required to comply with the 45-day minimum period and maintain records of disconnected numbers starting in July 2020 [88]. The RND became operational in November 2021 [89].

Currently, RND access is available to FCC-verified accounts for a fee; database users need to register as a caller, service provider, toll-free number administrator, or FCC personnel [90]. Other entities, however, may take steps to mitigate number recycling threats if given access to the database (e.g., a website may be able to use the RND to check reassigned numbers against SMS 2FA/recovery settings and warn users). In October 2021, we reached out to the FCC and suggested that it consider number recycling risks and to encourage RND access for websites and relying parties for this use case.

4.2.5. Structure of U.S. phone numbers

United States phone numbers are of the 10-digit format:

NPA-NXX-XXXX

NPA stands for Number Plan Area, or area code. There are currently 330 area codes in use in the U.S. NXX refers to the central office (exchange) code. In § 4.4, we take this structure into account in designing our sampling strategy.

4.2.6. Previous work on the risks of number recycling

There have occasionally been mentions of number recycling incidents in the media; one blog post had even speculated on the feasibility of taking over linked social media profiles with recycled phone numbers [91]. Recently, McDonald et al. conducted a user survey to ask 195 participants about their experiences with using phone numbers as identifiers and phone number recycling [81]. They determined these incidents occur regularly; many participants (72/195) reported experiencing negative downstream effects, such as receiving calls / texts meant for previous owners and being unable to add their number to online services due to an existing account.

These negative effects can be greatly amplified if exploited by an adversary. Our research is the first to analyze how adversaries can exploit phone number recycling with ease. To the best of our knowledge, there has not been any prior academic work looking at the wide scale security impact of number recycling. Specifically, none of the eight attacks we present in § 4.3 appear to have been systematically studied.

4.2.7. Related work

Beyond the effects of number recycling, SMS-based 2FA is less secure because it is tractable to known security weaknesses at mobile carriers. IMSI-catchers can be used to eavesdrop calls and texts by intercepting a nearby mobile phone's cell tower connection [26]. The signaling protocol used by carriers to achieve interoperability—Signaling System 7 (SS7)—does not authenticate requests, and thus can be used by remote attackers to re-route SMS 2FA messages to their own phones [31, 32]. Some carriers have weak (or weakly enforced) policies for authenticating subscribers over the phone (e.g., recall two recently dialed numbers); attackers can easily obtain this information and trick customer service representatives (CSRs) into updating the SIM card on a victim's account to one they control, in a SIM swap attack [1].

Some consumer email providers recycle usernames of dormant accounts. Like SMS-based authentication, email is commonly used to authenticate logins and recoveries. Of the top three providers, Yahoo and Microsoft both close accounts for inactivity and make the usernames available for new users [92, 93]. Google—the most popular provider—does not recycle email addresses [94]. While there has been significant backlash against Yahoo and Microsoft for prioritizing the ability to choose "short, sweet, and memorable" usernames over security and privacy, the practice remains unchanged. There has not been any analysis on the implications of recycling email addresses thus far [95, 96].

4.3. Overview of number recycling attacks

Table 4.1.: Eight attacks enabled by number recycling. We empirically investigated the feasibility of the three highlighted attacks.

Attack	Threat(s)	Population(s) affected
PII indexing. Attacker cycles through available numbers on the carrier's online number change form and checks for previous owners' personally identifiable information (PII) through people search services. They obtain the numbers that produce hits on these services.	Amass PII; create stepping stone to impersonate previous owner; read new messages intended for the victim	Previous owners; friends and family of previous owners
Account hijackings via recovery. Attacker cycles through available numbers and checks if any of them are linked to existing online accounts (e.g., social media, email, e-commerce). They obtain the numbers with hits and try to reset the password on the linked accounts via SMS-based password recovery.	Hijack online accounts; impersonate previous owner; read new messages intended for the victim	Previous owners; friends and family of previous owners

Continued on next page

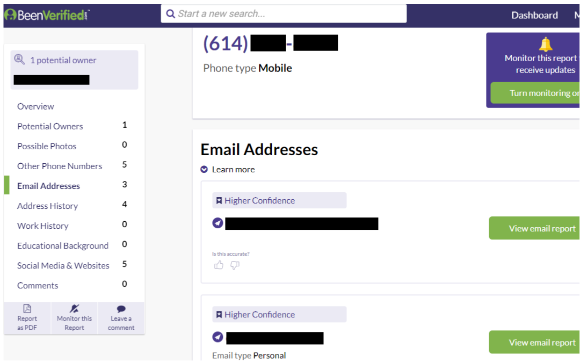
Table 4.1.: *(Continued)* Eight attacks enabled by number recycling. We empirically investigated the feasibility of the three highlighted attacks.

Attack	Threat(s)	Population(s) affected
Account hijackings without password reset. Attacker cycles through available numbers and checks for linked accounts as well as previous owner PII on people search services. Attacker uses the PII to find and purchase passwords from data breach listings on cybercriminal marketplaces. They obtain the phone numbers that are linked both to online accounts and to breached passwords. They bypass SMS-based 2FA on the online accounts using the password and control of the phone number.	Hijack online accounts even with SMS 2FA enabled; impersonate previous owner; read new messages intended for the victim	Previous owners; friends and family of previous owners
Targeted takeover. Attacker learns that an acquaintance's contact has changed (e.g., stalker calls and gets a cancelled number intercept message, friend changes their number and tells everyone). They keep track of the aging period, and obtain the number once it becomes available.	Hijack online accounts; impersonate/stalk previous owner; read new messages intended for the previous owner	Previous owners, especially intimate partner violence (IPV) survivors changing their numbers to escape abusers
Phishing. Attacker logs available numbers but does not obtain them. Later, they keep checking whether the numbers are still available. Once a number is assigned to a new subscriber, they can phish the subscriber through SMS (e.g., "Welcome to your new service. Click here to enable high-speed data for your account"). Subscribers are more likely to fall for phishing attacks when the message sounds believable [97].	Hijack victims' online phone accounts; potentially take control of victims' phone numbers.	Subscribers who have been assigned a new number, whether fresh or recycled.
Persuasive takeover. Attacker logs available numbers but does not obtain them. After the number is assigned, they can spoof a carrier message (e.g., "Your number is part of an ongoing investigation on the previous owner and needs to be reclaimed. Please change your number online") and obtain the number for himself after the aging period.	Hijack online accounts with phone number linked; impersonate victim; read new messages intended for the victim	Subscribers who have been assigned a new number, whether fresh or recycled.

Continued on next page

Table 4.1.: *(Continued)* Eight attacks enabled by number recycling. We empirically investigated the feasibility of the three highlighted attacks.

Attack	Threat(s)	Population(s) affected
Spam. Attacker obtains a number, intentionally sign up for various alerts, newsletters, campaigns, and robocalls, and then release the number for recycling	Victim harassed with unwanted texts and calls; account calling balance depleted	Subscribers who have been assigned a recycled number.
Denial of service. Attacker obtains a number, sign-up for an online service that requires a phone number, and releases the number. When a victim obtains the number and tries to sign up for the same service, they will be denied due to an existing account. The attacker can contact the victim through SMS and demand payment to free up the number on the platform.	Denial of service; victim needs to pay ransom to use platform	Subscribers who have been assigned a recycled number and are new users of online services that require a unique phone number.



[redacted]

(614) [redacted]-[redacted]

Phone type **Mobile**

Email Addresses

Higher Confidence

[redacted]

Higher Confidence

[redacted]

Email type Personal

Figure 4.1.: Anyone can enter a phone number on BeenVerified to reveal personally identifiable information (PII) on the number's previous and current owners.

We present the first systematic analysis of number recycling attacks. In Table 4.1, we present eight different threats enabled by number recycling, four in which attackers can target previous owners of recycled phone numbers, and four in which attackers can target future owners.

Number recycling can be leveraged in different attacks ranging from opportunistic to highly targeted. We selected the first three attacks in Table 4.1 to study in depth because they are both serious and can be studied without harming actual subscribers. We now describe them in more detail.

In an opportunistic scenario with the lowest barrier to entry, an attacker can use a recycled number—that they have obtained by signing up for service—to look up information on the number's previous owner on the web or through data aggregation services, which are available to anyone at low cost (**PII indexing**). Fig. 4.1 shows lookup results at one such service, BeenVerified; a report can include information like previous owner names, photos, email addresses, work history, social media account handles. Armed with personally identifiable information (PII) and control of the number, the attacker can impersonate previous owners in calls and messages.

Consider another scenario: an attacker can use the recycled number to look for and break into linked profiles online via SMS-authenticated password resets (**Account hijackings via recovery**). Despite growing awareness of the risks of SMS-based authentication of online accounts, the practice remains prevalent [1].

Alternatively, the attacker can find and use the previous owner's email addresses to look for password breaches and purchase the stolen password on the dark web.[10][11] With the stolen password, the attacker can log in to most of the previous owner's accounts without going through recovery, and defeat SMS 2FA by receiving the passcode sent to the recycled number (**Account hijackings without password reset**). Note that the recovery pages usually don't reveal PII such as email addresses (only the existence of an account and available recovery methods), so the attacker needs to use **PII indexing** as a gateway to this attack.

[10] PII—usually email addresses—are often used as usernames.

[11] Most users are known to notoriously practice poor security hygiene by reusing their passwords, so a purchased password may work at multiple websites.

An adversary might not even need to obtain the phone number in order to plan out an attack. At carriers that allow for full numbers to be previewed—either during signup or number change—an attacker can "scout out" a number by looking for linked accounts and owner history, all before obtaining the recycled number. As we will show later, this strategy is made possible by the lack of query limits on the carrier interfaces in our study (§ 4.6.1).

Attackers may have varying economic motivations for these attacks [98]. They may be interested in stealing money from victims, such as by taking over online accounts that hold cryptocurrency [12]. Alternatively, they may use amassed accounts on social media for spam campaigns or fake followers [99, 100]. The latter strategy requires a relatively large number of online accounts, and a correspondingly large number of phone number changes (assuming that the attacker controls a fixed number of SIM cards and service plans). Unfortunately, at the time of our study, some carriers not only had no query limits in place but also no rate limits for phone number changes (§ 4.6.1).

In our study, we simulated an opportunistic attacker with access to data aggregation (people search) services, data breach lookup tools, and one prepaid account per carrier, all of which can be obtained for under $100. We did not target any specific area codes, and we did not look for vulnerabilities before "obtaining" (logging) the numbers.

Note that our attacker is a *UI-bound adversary*—an authenticated user who uses the system with the same privileges as any other user, albeit with malicious intent [21]. Since the adversary operates within the functionality of the user interface and does not need to use any tools or exploit a system vulnerability, the population of potential attackers is expansive.

IPV survivors are especially vulnerable to targeted takeovers. Survivors of intimate partner violence (IPV) face a higher risk of harm from number recycling attacks. Survivors may change phone numbers to escape their abusers [101]. Upon realizing that their victim's number has changed, the abuser (a *UI-bound adversary*) may keep track of the aging period and obtain the number once it becomes available (**Targeted takeover**). Armed with access to the survivor's old number and PII, as well as a desire to agonize, the abuser can cause devastating harm. For example, the abuser can hijack online accounts where the survivor has either forgotten or has not yet updated the SMS 2FA and recovery number. The abuser may also be able to impersonate the survivor via SMS to manipulate mutual acquaintances (e.g., trick friends into revealing the survivor's current number, or convince them that the survivor is no longer being stalked). Since they have already moved on to using a new number, survivors may be unaware that their abuser is using their previous number.

4.4. Analysis of attacks against previous owners

We study the severity of the security risks associated with phone number recycling, and find that previous owners of most recycled numbers are at risk.

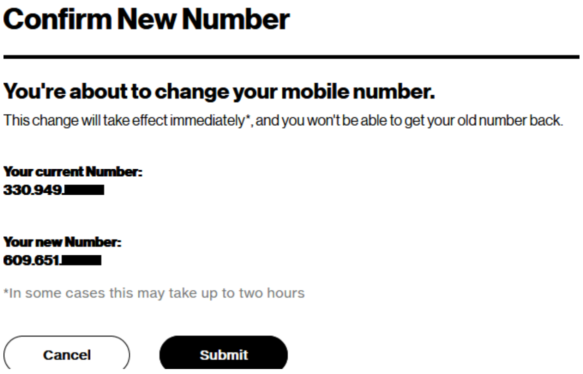
4.4.1. Method

We aim to answer three questions:

1. How easily can attackers find recycled phone numbers and corresponding PII on their previous owners?
2. How easily can attackers find recycled phone numbers with vulnerable linked online accounts?

3. Is it feasible for attackers to use PII from people search sites to look for likely passwords for these linked accounts?

4.4.1.1. Sampling available prefixes and numbers



Confirm New Number

You're about to change your mobile number.

This change will take effect immediately*, and you won't be able to get your old number back.

Your current Number:
330.949.[redacted]

Your new Number:
609.651.[redacted]

*In some cases this may take up to two hours

Figure 4.2.: Verizon's number change interface for prepaid subscribers.

We signed up for one prepaid account at each of the two largest U.S. carriers—Verizon Wireless and T-Mobile. Both carriers provide an online interface for subscribers to change their phone number. The third major carrier—AT&T—does not, so we omitted it from our study. We manually interacted with the interfaces just as a normal subscriber looking to change their number would. Throughout, we logged available numbers but did not complete any number changes.

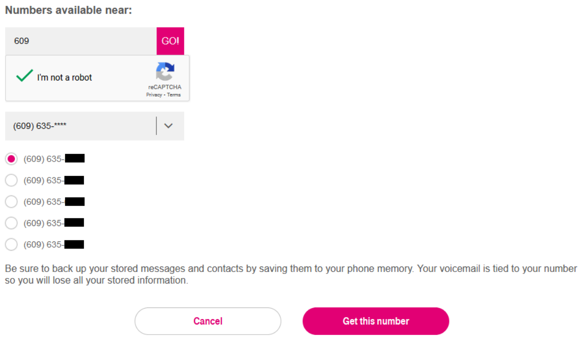
All of the number change interfaces we saw in this study index available numbers by NPA-NXX prefixes; that is, subscribers need to choose an available NPA-NXX as an intermediate step. This constraint affects our number sampling strategy. At Verizon, we were able to randomly sample prefixes, but not numbers. We were unable to randomly sample prefixes at T-Mobile due to further selection constraints we highlight later in this section.

Verizon. Verizon allows prepaid subscribers to specify any NPA-NXX as criteria on the online number change request form. If the entered NPA-NXX is a valid Verizon prefix with at least one available number, the following screen will denote a single selected number with a predefined subscriber number (last 4 digits, see Fig. 4.2). The subscriber can either confirm the request (after which their line will be updated, often immediately) or go back to perform a new query. If the subscriber performs a new query with the same NPA-NXX, the following screen will show a different number from the previous query results. If the entered NPA-NXX is not serviced by Verizon or currently has no available numbers, the subscriber is presented with an error modal asking for a valid NPA-NXX entry. Since we also encounter the error modal at different iterations of repeated queries for each NPA-NXX, we assume that the system temporarily keeps track of "seen" numbers and errors out when we have exhausted the available number pool for each prefix.

We started with a list of all currently active NPA-NXX prefixes by obtaining the central office code assignment records hosted on NANPA.[12] At the time of our experiment, there were 180,741 unique prefixes on record, and thus in use by telecoms in the U.S. We randomly selected prefixes, and for each prefix, we leveraged the number change request form to log all available numbers. That is, we repeatedly requested a new number with the same prefix until we encountered the invalid NPA-NXX message, and continued the process for all NPA-NXX prefixes in our list. We iterated 875 prefixes over the course of three days, for a total of 8,603 available numbers across 77 of those prefixes. The largest prefix contained over 900 numbers, while there were 28 prefixes with under 10 available numbers.

T-Mobile. T-Mobile allows prepaid subscribers to specify any NPA as a query on the online number change request form. The system returns up to five NPA-NXX with the most available numbers (the raw JSON response contains an inventory count for each NXX). For each of the five NPA-NXX's, five available numbers are shown for the subscriber to choose from, for a maximum of 25 numbers per NPA

[12] https://nationalnanpa.com/reports/reports_cocodes_assign.html. (08/16/2020).



Numbers available near:

609 GO!

✓ I'm not a robot

(609) 635-****

● (609) 635-[redacted]

○ (609) 635-[redacted]

○ (609) 635-[redacted]

○ (609) 635-[redacted]

○ (609) 635-[redacted]

Be sure to back up your stored messages and contacts by saving them to your phone memory. Your voicemail is tied to your number so you will lose all your stored information.

Figure 4.3.: T-Mobile's number change interface for prepaid subscribers.

(Fig. 4.3). Barring churn from other subscribers' activities, the 25 numbers do not change between subsequent queries. We iterated through the 330 active area codes and leveraged the number change request form to log accessible available numbers. We collected 6,928 available numbers across 1,393 NPA-NXX prefixes.

4.4.1.2. Identifying likely recycled numbers

In the next step, we focused on recycled numbers. We simulated an adversary trying to maximize chances of finding a recycled number. Accordingly, for both carriers, we restricted our attention to NPA-NXX blocks for which no two available numbers were within 10 of each other. Since new NPA-NXX blocks are more likely to have consecutive available numbers (like how newly printed money is consecutively numbered in stacks), an adversary who is interested in recycled numbers can ignore those blocks in their queries.

We therefore grouped the blocks into two categories:

- ***Likely recycled.*** No two available numbers are within 10 of each other. Numbers from this pool are likely to have been previously assigned.
- ***Possibly unused.*** At least two numbers are within 10 of each other. The pool consists of both unused numbers and some recycled numbers that are close together just by chance.

Table 4.2.: A detailed breakdown of applying our number classification strategy.

(a) T-Mobile

	Available Numbers	NPA-NXXs
Likely recycled	1,438	295
Possibly unused	5,490	1,098

(b) Verizon

	Available Numbers	NPA-NXXs
Likely recycled	159	32
Possibly unused	8,444	45

Table 4.2 details the result of splitting the NPA-NXX blocks along the constraint. For Verizon, it may seem that *Likely recycled* numbers are rare in comparison to *Possibly unused* numbers. However, the number of NPA-NXX blocks in each group are actually comparable; if a Verizon subscriber selects a NPA-NXX at random they can happen upon a *Likely recycled* number nearly half of the time. Furthermore, numbers from the *Possibly unused* group can also be recycled. At T-Mobile, we logged nearly four times as many NPA-NXX blocks from the *Possibly unused* group as blocks from the *Likely recycled* group. This is possibly due to T-Mobile's interface design; NPA-NXX blocks with the most available numbers are most likely new blocks, and therefore appear in the five NPA-NXX choices more often.

4.4.1.3. Reverse lookups

For each of the 159 numbers in Verizon's *Likely recycled* group and 100 randomly sampled numbers in T-Mobile's *Likely recycled* group, we used the reverse phone lookup tools at two people search services—BeenVerified and Intelius—to look for owner history. We chose these two services based on positive user reviews [102, 103]. This step serves two purposes. It allows us to estimate the vulnerability to the **PII indexing attack** (§ 4.4.2). It also lets us validate our strategy for classifying numbers as *Likely recycled* and *Possibly unused*. We did so by randomly sampling 159 and 100 numbers from Verizon's and T-Mobile's *Possibly unused* groups respectively and looking for people search hits. We found that 53/159 and 44/100 of the sampled *Possibly unused* numbers returned hits, compared to 96/159 and 75/100 of the sampled *Likely recycled* numbers. For each carrier, we used a one-sided z-test to evaluate if these difference was significant, and we found strong support for the hypothesis that the hit rate in the *Likely recycled* group was greater than that of the *Possibly unused* group ($p < 0.0001$ for both carriers).

In addition to finding hits, we also logged any associated email address that appeared in the owner history. For each address, we checked for involved password breaches on *Have I Been Pwned?* (HIBP)—an online service that allows users to check whether their credentials and other identifying information have been compromised in data breaches. This enabled us to quantify the effectiveness of the **account hijacking without password reset** attack (§ 4.4.2).

Finally, we measured the fraction of *Likely recycled* numbers linked to existing online profiles. For each number in the sample, we used the account recovery feature of Amazon, AOL, Facebook, Google, Paypal, and Yahoo to locate any linked accounts, as an adversary would. In contrast to an adversary, upon receiving a response (account found/not found), we aborted the recovery process. The procedure allowed us to determine whether an available number was still linked to an existing account. We selected Google (Alexa Rank 1; Google's YouTube is AR 2), Amazon (AR 3), Yahoo (AR 4), and Facebook (AR 5) based on their popularity in the U.S. We selected Amazon, AOL, Paypal, and Yahoo because they allow simultaneous use of SMS 2FA and SMS account recovery on new (previously unseen) devices, which was found in a previous study looking at SIM swaps [1]. Accounts with this *doubly insecure* configuration—a term coined by the study which we borrow for the remainder of our paper—are at immediate risk of takeover, an adversary can hijack a linked account just by obtaining a recycled phone number. These websites remain *doubly insecure* as of August 2020.[13] The other two websites in our study—Google and Facebook—use SMS-based recovery conditional on 2FA settings; SMS recovery is allowed only if SMS 2FA is not enabled. This enabled us to quantify the effectiveness of the **account hijacking via recovery** attack (§ 4.4.2). We were aided by the fact that all websites we selected give a negative response if no linked account is found.

4.4.1.4. Ethical considerations and responsible disclosure

We registered our method with our university's Institutional Review Board in July 2020. Our research plan was ruled as non-human subjects research. Nevertheless, we took steps to mitigate the risk of harm to previous owners of the phone numbers in our study. We determined—through our own accounts—that initiating account recovery with a phone number and aborting once a linked account is found does not raise any alerts to the user at any of the six services studied. Secondly, we deleted all identifying information (e.g., phone numbers, emails) at the end of our study. Lastly, we kept the *Likely recycled* numbers in our study relatively small as to avoid any erroneous overshoots in account recovery processes, which we executed manually.

We performed these measurements in August and September 2020, and provided initial notification to the carriers we studied and CTIA on October 22, 2020. We presented our findings to major carriers and CTIA in November 2020.

[13] We verified the *doubly insecure* configuration on newly-created accounts with no associated assets on two different devices. It is possible that these websites employ additional authentication for real-world accounts based on activity or some other notion of value.

4.4.2. Results: previous owners of most recycled numbers are at risk

Table 4.3.: Hit rates from our testing methods. Most of the numbers we analyzed were confirmed recycled (83%). Rows highlighted in yellow suggest immediate danger to accounts with a certain authentication configuration. Rows highlighted in red suggest immediate danger to accounts, regardless of authentication configuration.

Test	Attack	Hit count: T-Mobile (out of 100)	Hit count: Verizon (out of 159)	Hit count: total (out of 259)
Found on people search services **OR** linked account at any of the six websites	Confirm that number is recycled	94 (94%)	121 (76%)	215 (83%)
Found on people search services	PII indexing	75 (75%)	96 (60%)	171 (66%)
Linked account at any of the six websites	Account hijackings via recovery (if SMS-based recovery is enabled)	79 (79%)	92 (58%)	171 (66%)
Linked account at any of the four *doubly insecure* websites	Account hijackings via recovery	44 (44%)	56 (35%)	100 (39%)
Amazon	Account hijackings via recovery	17 (17%)	17 (11%)	34 (13%)
AOL	Account hijackings via recovery	4 (4%)	5 (3%)	9 (3%)
PayPal	Account hijackings via recovery	16 (16%)	19 (12%)	35 (14%)
Yahoo	Account hijackings via recovery	22 (22%)	43 (27%)	65 (25%)
Linked account at any of the six websites **AND** involved in a password breach	Account hijackings without password reset	50 (50%)	50 (31%)	100 (39%)

We document the hit rates of our testing methods on all 259 numbers in Table 4.3. As mentioned in § 4.3, each method to test was motivated by a corresponding attack—presented in Table 4.1—that an adversary can leverage on previous owners upon taking control of the number.

Our findings are as follows:

1. **Most numbers enable impersonation attacks through *PII indexing*.** Of the 259 numbers we analyzed, 171 (66%) produced a hit at either BeenVerified or Intelius. As previously described, an attacker can use these services to gather previous owners' PII. Once they obtain the previous owner's number, they can perform impersonation attacks.

2. **Most numbers enable *account hijackings via recovery*.** 171 / 259 numbers in our sample (66%) had a linked existing account on *at least* one of the six websites. An attacker can potentially break into all of these accounts—even at Facebook and Google if SMS-based recovery is enabled (highlighted yellow in Table 4.3).

   One especially concerning result is the hit rate at *doubly insecure* websites: Amazon, Yahoo, Paypal, and AOL. 100 (39%) of the numbers we sampled had a linked account on ***at least*** one of the four websites (highlighted red in Table 4.3)

   We do not know how many of the accounts in our sample had SMS-based recovery enabled since we aborted the account recovery process after determining whether a linked account exists. However, for a subset of numbers—68 of 171 (26%)—we can confirm that the accounts are definitely vulnerable. These numbers were linked to accounts at Yahoo or AOL, both of which have no alternative to *doubly insecure* configurations (Amazon and Paypal do have secure alternate configurations, though not by default).

3. **Some numbers enable *account hijackings without password reset*.** In total, we found 100 phone numbers (39% of our sample) with at least one associated email address that had been involved in a password breach and had linked profiles on at least one of the six websites. Apart from the *doubly insecure* sites, the rest of the websites in our analysis (Facebook and Google) allow for

SMS 2FA, and thus are as vulnerable to this attack as much as the other four (highlighted yellow in Table 4.3).

4. **Other authentication methods are also at risk of takeover.** Three of the six websites we analyzed—Google, Yahoo, and AOL—provide consumer webmail services in the U.S. 139 of the 259 numbers (54%) were linked to an account on at least one of the three websites. As a common recovery and 2FA option, email-based passcodes can also be intercepted once an attacker hijacks the inbox with a recycled phone number.

Our key finding is that attackers can feasibly leverage number recycling to target previous owners and their accounts. The moderate to high hit rates of our testing methods indicate that most recycled numbers are vulnerable to these attacks. Furthermore, by focusing on blocks of *Likely recycled* numbers, an attacker can easily discover available recycled numbers, each of which then becomes a potential target.

4.5. Analysis: inventory of recycled numbers

According to the FCC, 35 million phone numbers in the U.S. are disconnected each year [41]. This suggests that a vast number of recycled numbers may be available to attackers. In this section, we quantify the inventory of recycled numbers in two steps: first we analyze a snapshot in time; then we analyze the churn rate. We confirm that a large number of recycled numbers (about one million) are available at Verizon, and tentatively find that this inventory of recycled numbers is largely replaced by a fresh set of numbers within a month.[14]

[14] We are unable to estimate the corresponding numbers for T-Mobile due to restrictions of the online interface that prevented us from viewing all available numbers.

4.5.1. Recycled numbers estimates

We used the following strategy for estimating the number of available recycled numbers at Verizon.

- Let P be the number of all available phone numbers.
- Let R be the number of all available phone numbers that are recycled. This is our estimand.
- Let r be the probability that a number selected is recycled. By definition, $r = \frac{R}{P}$
- Let S be the number of numbers from NPA-NXX blocks with no two available numbers being within 10 of each other. **We assume that all such numbers are recycled.**
- Let H be the hit rate at people search services; that is, the proportion of numbers that return any information on past owners.
- By our assumption, $H_R = H_S$

$$\begin{aligned}
H_P &= \frac{R}{P} H_R + (1 - \frac{R}{P}) H_{\bar{R}} && \text{by definition} \\
&= \frac{R}{P} H_R && \text{We set } H_{\bar{R}} \text{ to 0 since a new number won't get any hits} \\
&= \frac{R}{P} H_S && \text{by substitution} \\
&= r\, H_S && \text{by substitution} \\
r &= \frac{H_P}{H_S}
\end{aligned}$$

We now have two expressions for r; equating them, we get $R = P\frac{H_P}{H_s}$. Our measurements allowed us to estimate each of the three quantities on the right hand side of this equation as follows.

To estimate P (Verizon's inventory of available numbers), we extrapolated the results of our iteration through available NPA-NXXs in § 4.4. We had exhaustively iterated 875 of the valid NPA-NXX prefixes and logged 8,603 available numbers. Since there are 180,741 valid NPA-NXX prefixes, we estimate P to be 1.8M (95% CI [860K, 2.7M]).

In our lookups at people search services in § 4.4.1.3, we had found H_S to be 96/159, and the hit rate from the *Possibly unused* pool to be 53/159. We then computed H_P by taking a weighted sum of those two sample proportions. We estimate R—the available number of recycled numbers—to be 996K (95% CI [420K, 1.6M]).

Recall that in the previous section we simulated an adversary trying to maximize chances of finding a recycled number. He restricts himself to the *Likely recycled* pool—NPA-NXX blocks for which no two available numbers were within 10 of each other. Even with this restricted strategy, the number of available recycled numbers at any given time is vast: we estimate S to be 33K (95% CI [18K, 48K]).

While the total number of available recycled numbers is important in terms of an adversary seeking to carry out large-scale attacks, the probability of receiving a recycled number from navigating the online interface is also relevant since it quantifies the risk to a subscriber seeking a fresh number. If a Verizon prepaid subscriber were to change their number online by entering an NPA-NXX at random, she would receive a recycled phone number 41.6% of the time (95% CI [30.5%, 52.6%]). This figure assumes all *Likely recycled* numbers are recycled, and that all *Possibly unused* numbers are brand new.

4.5.2. Churn analysis

New recycled numbers become available over time, in accordance with FCC number aging rules. To quantify number churn at Verizon, we randomly selected 20 of the

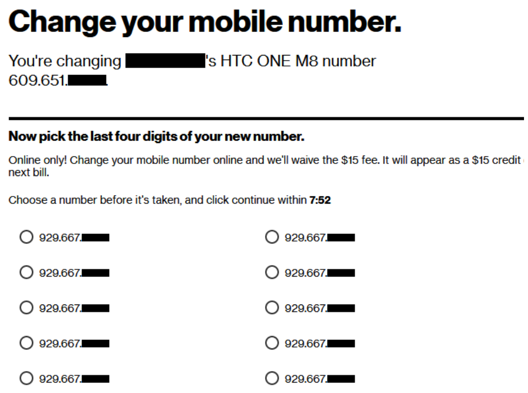
77 NPA-NXXs from our initial collection phase (§ 4.4.1.1) and logged all available numbers. 15 of the 20 selected NPA-NXXs had availability in September. We collected numbers at the end of September and October 2020.

We made two key findings:

1. **Available numbers are assigned quickly.** We measured churn by dividing the size of inventory lost at the end of the month (numbers that do not appear in the next month's dataset) by the inventory size at the beginning of the month. We estimate the monthly number churn rate to be 86.5% (95% CI [85.2%, 87.8%]); only 330 of the 2,449 total logged numbers in September were still available in October. Assuming a constant monthly churn rate, we estimate that an available number gets taken after 1.2 months. Individually, most NPA-NXXs had high monthly turnover. Of the 15 NPA-NXXs, 12 of them had at least 80% churn during the month of observation, eight NPA-NXXs had a 100% churn rate during observation. Only two NPA-NXXs had churn rates below 50%; we speculate these are prefixes in areas with numerous other highly available prefixes (since the number change interface allows geographic queries as well) or in areas with little subscriber activity.
2. **New recycled numbers were being made available over time.** Six of the eight NPA-NXXs had new available numbers that resembled *Likely recycled* traits (i.e., no two available numbers are within 10 of each other).

Taken together, these findings suggest that not only are about one million recycled numbers available at any one time (§ 4.5.1), but also that a largely fresh set of recycled numbers becomes available within one month.

Unfortunately, we were unable to analyze churn after October. On November 17, 2020, we discovered Verizon had patched their prepaid backend system to return only a limited set of available numbers for each NPA-NXX, although we could still



Change your mobile number.

You're changing [redacted]'s HTC ONE M8 number 609.651.[redacted]

Now pick the last four digits of your new number.

Online only! Change your mobile number online and we'll waive the $15 fee. It will appear as a $15 credit next bill.

Choose a number before it's taken, and click continue within **7:52**

○ 929.667.[redacted] ○ 929.667.[redacted]

○ 929.667.[redacted] ○ 929.667.[redacted]

○ 929.667.[redacted] ○ 929.667.[redacted]

○ 929.667.[redacted] ○ 929.667.[redacted]

○ 929.667.[redacted] ○ 929.667.[redacted]

Figure 4.4.: Verizon's number change interface for postpaid subscribers. We have redacted the last four digits of each number.

make unlimited queries. As such, we were unable to measure longitudinal trends of Verizon's numbering resources.

4.6. Analysis of carrier interfaces and recycling policies

4.6.1. Most number change interfaces have no limits

Adversaries can take advantage of the lack of limits on number change interfaces to quickly discover recycled numbers and carry out attacks. We further investigated the interfaces at T-Mobile and Verizon for postpaid and prepaid subscribers. Using carrier-published FAQs, webpage element inspection, and interactions with the interface (including interactions from § 4.4.1.1), we documented the change and query limits carriers had in place. Our findings are shown in Table 4.4.

Both T-Mobile and Verizon prepaid interfaces allow for unlimited queries on available numbers. T-Mobile additionally does not place limits on changes. Both carriers impose limits on their postpaid subscribers: Verizon limits both the number queries and amount of changes, while T-Mobile does not support online number

Table 4.4.: Characteristics of the online number change interfaces at T-Mobile and Verizon for prepaid / postpaid subscribers.

	T-Mobile		**Verizon**	
	Prepaid	**Postpaid**	**Prepaid**	**Postpaid**
Change limit(s)	None	Online number changes are not supported; changes can only be done over the phone by calling customer service	3 changes per day; 5 changes per rolling 30 days	1 change every 7 days
Query limit(s)	No limit on amount of queries; up to 5 NXXs per NPA query, up to 5 available numbers per NXX (25 total numbers per NPA)		Queries not allowed if there are any change limits in effect, otherwise, no limits	6 NPA-NXX queries / day; up to 10 available numbers shown per NPA-NXX; subscriber is allowed 10 minutes to select an available number; queries not allowed if change limits reached
Fee(s)	Free	1 free change per year, per line; additional changes $15	Free	Free if done online

changes. All online interfaces display full numbers, which gives an attacker the ability to discover recycled numbers before confirming a number change.

Despite having more limits on their online interfaces (or lack of an interface altogether), postpaid customers are not immune to number recycling threats. We discovered both carriers using the same number pools when we were able to change the number on our postpaid lines (T-Mobile postpaid over the phone) to numbers we had seen on their prepaid interfaces. This means that **postpaid subscribers are also at risk for number recycling attacks**, despite throttling in their interfaces. In fact, attackers may choose to use prepaid accounts due to lower cost and absence of identity checks.

4.6.2. CSRs had inconsistent responses about aging periods

In addition to investigating interfaces, we attempted to learn the number recycling policies at T-Mobile and Verizon. Since neither carrier offers public-facing documentation on the matter, we called CSRs at each carrier and inquired about the status of our old numbers in a number change, using a different account each time. We asked for the aging period—the time between subscribers losing access to their old number and the number being available for assignment. As mentioned, the FCC-mandated minimum aging period is 45 days (§ 4.2). We placed 13 calls at each carrier—ten at prepaid and three at postpaid—from September to November 2020.

We found that CSR responses were wildly inconsistent.

1. At T-Mobile, we received seven different responses across 13 calls.
2. At Verizon, we received eight different responses across 13 calls.
3. Responses were highly varied. The purported aging period ranged from one hour to one year at T-Mobile, and one week to four months at Verizon.
4. At both carriers, there was no majority response, however, the plurality response at each was 30 days.
5. In two instances at each carrier, CSRs mentioned there was no specified aging period policy. In one of those instances at Verizon, the CSR purported that all previous numbers remained linked to the account—and could not be reassigned—as long as the account remained active.

Based on the widely different responses we received, we were unable to determine either carrier's current recycling policies. Furthermore, the inconsistent knowledge among company personnel also poses a concrete problem for subscribers.

4.6.3. Subscriber confusion about carrier recycling practices could result in security issues

If CSRs at T-Mobile and Verizon are uninformed of number recycling policies, they may end up passing incorrect information to subscribers. We systematically searched carrier-hosted community support forums at all major carriers—AT&T, T-Mobile, and Verizon—by using number recycling-related queries, and noting responses on top relevant posts as of January 2021. We further examined independent forums by searching with the same querystrings along with the carrier name. We noted nine different responses across seven posts.

On both types of forums, speculation on aging periods varies widely; responses ranged from no aging period to six months. Four responses claimed numbers were reassigned in 60 days, and three responses—one from company staff—claimed that numbers were reassigned in six months (see § B.1 for individual subscribers' statements).

These responses should be interpreted anecdotally, primarily due to limited number of posts and responses we were able to find. Regardless, the lack of any public-facing documentation and inconsistent CSR knowledge exacerbate the problem. Subscriber uncertainty about number recycling can have serious security consequences. Previous owners may incorrectly perceive the aging period for their disconnected numbers to be much longer than it actually is, and put off updating their online accounts. In the meantime, those numbers may have become available again for other subscribers—possibly attackers—to obtain. Temporarily-disconnected subscribers are also affected: they may return to find that their number has been reassigned, despite being told of a longer aging period.

4.7. Analysis of calls and texts meant for previous owners of recycled numbers

So far, our analysis has centered on a motivated adversary who is aware of number recycling vulnerabilities and exploits them via online number change interfaces. Now we consider the perspective of a subscriber who is unknowingly assigned a recycled number and opportunistically exploits vulnerabilities.

We seek to estimate the fraction of recycled numbers which receive sensitive communications meant for previous owners without the need for any explicit action by the new subscriber. Such messages may by themselves compromise the privacy of the previous subscriber or alert the new subscriber to the fact that they are in a position to exploit a security vulnerability.

4.7.1. Method

4.7.1.1. We obtained 200 recycled numbers

At T-Mobile and Verizon, we signed up for 10 prepaid accounts, for a total of 20 accounts. For each Verizon account, we entered a random NPA-NXX and checked if the returned available number was linked to accounts at any of the six websites we studied in § 4.4.1.3. If so, we confirmed the change and obtained the recycled number, otherwise, we randomly selected a new NPA-NXX and repeated the process. Similarly, at each T-Mobile account, we entered a random NPA and iteratively looked up the 25 selectable numbers (interface details in § 4.4.1.1) until we found and obtained one with a linked online profile. We repeated the process at all 20 accounts for 10 weeks, giving us a total of 200 recycled phone numbers that we monitored for one week each.

4.7.1.2. We collected info about incoming calls / texts

We kept all 20 phone accounts powered-on and actively connected for the entire 10-week period while monitoring incoming calls and messages. All accounts were provisioned on unlocked Android phones. We restarted the devices only after a number change each week. At the end of each week, we ran an Android application to 1) write the timestamp, sender phone number, and communication type (call / text) to a file on device storage, and 2) clear the call log and message inbox. We retrieved the file onto our computer and used it in our analysis.

We ran our honeypot from November 2020 to January 2021, and received 1491 total calls / texts (561 texts, 930 calls) from 1064 different senders. It is important to note that these unsolicited personal calls / texts made to our honeypot should mainly be the result of number recycling, but in rare cases, they can be the result of an incorrectly dialed number.

4.7.1.3. Identifying sensitive calls / texts with only metadata

To identify sensitive calls, we collaborated with Nomorobo—a robocall blocking service. We selected Nomorobo because of its popularity in the robocall detection space and its recent collaboration with academic researchers in a longitudinal study on robocalls [104]. We worked directly with the company's founder, who used Nomorobo's honeypot data to identify spam robocalls and likely spoofed numbers in our dataset. We were also provided with an allowlist of callers; that is, legitimate robocalls that appeared in our dataset. From the allowlist, we were able to infer the nature of the calls we received.

To identify sensitive messages, we focused on short codes—5-6 digit phone numbers—seen in our dataset. Short codes—which are used to send high-throughput content, such as marketing, alerts, and 2FA messages—are regulated differently from 10-digit numbers, making them harder to spoof and easier to find owner (organization) information.[15] We manually classified the 48 seen short codes in our dataset by looking up their owner in the publicly-available owner database, by texting “HELP”—a standardized keyword to request service information—from our personal phone numbers, and by searching the web for websites that mention this short code.

4.7.1.4. Ethical considerations

We registered our method with our university’s IRB in July 2020. Our research plan was ruled as non-human subjects research. We also checked to make sure there were no legal issues with receiving communications meant for previous owners. Nevertheless, we took steps to mitigate the risk of harm to previous owners of the phone numbers in our honeypot. As we did in our analysis of attacks against previous owners (§ 4.4), we determined that reverse lookups on all six websites did not raise alerts to the previous owner. Secondly, we deleted all collected data at the end of the study. Most importantly, we took steps to protect previous owners’ privacy: we only collected call / text metadata. We developed an app to collect metadata and clear out inboxes and call logs, ensuring no member of the research team would need to view message content. We made this decision despite knowing that it could result in underreporting the number of sensitive messages.

Table 4.5.: A breakdown of identified calls / texts. We inferred the nature of communication using metadata.

Nature of call / text	Unique senders	Total calls / texts	Recycled numbers affected (out of 200)
Security/privacy-sensitive	24	60	19 (9.5%)
Authentication OTPs	7	13	6 (3%)
PII	17	47	14 (7%)
Marketing	19	40	13 (6.5%)

4.7.2. Results: nearly 10% of numbers received sensitive calls / texts meant for previous owners

We documented the number of sensitive calls / texts sent to our honeypot in Table 4.5. Our findings are as follows:

1. **19 numbers in our honeypot—nearly 10%—received sensitive calls / texts meant for previous owners.** These numbers received calls / texts containing PII or authentication passcodes. Upon receiving sensitive communication meant for the previous owner, a subscriber can realize his exploitative position and target the previous owner and her accounts. We highlight that this was the result of just one week of monitoring; it is possible that we could have identified more messages (and vulnerable numbers) if we monitored for longer.

    a) **6 numbers were still getting authentication calls / texts.** We identified seven senders that were associated with 2FA passcodes: Apple, Cash App, Facebook, Google, Microsoft, and WhatsApp (2 different numbers). As a result of losing their number, previous owners are now locked out of their accounts since they are unable to receive the sent OTP. Additionally,

[15] CTIA oversees short code assignments and maintains an owner database that is publicly available.

the adversary—after seeing the call / text—can zero in on hijacking the previous owner's account because he has now learned that she 1) has a linked account, and 2) uses SMS authentication, which he can defeat.

   b) **14 numbers received PII-revealing calls / texts.** We identified 17 senders that were associated with PII-revealing messages. These included pharmacy calls, school alerts, hospital calls, appointment reminders, and mobile banking texts. These potentially contain PII, which the adversary can amass to threaten previous owners. Worse, the adversary can possibly manipulate appointments and prescriptions by responding.

2. **Separately, 13 numbers received unsolicited marketing texts.** Apart from our main finding, we identified 19 short codes owned by marketing campaigns, totalling 40 texts. Yet we did not consent to receiving these messages. This demonstrates a known issue faced by marketing campaigns: under the Telephone Consumer Protection Act (TCPA), subscribers must opt-in to receiving messages, however, the senders currently have no practical way of determining changes in number ownership.[16] It is important to note that marketing campaigns may apply to use the RND, so the number of unsolicited marketing messages may decrease in the future.

Our key finding is that a significant proportion of our obtained recycled numbers still received sensitive communication during their one-week monitoring period. Through industry collaboration and short code lookups, we were able to use only metadata to infer the nature of received calls / texts in our honeypot, and conservatively quantify a direct consequence of subscriber confusion about number recycling.

[16] 47 U.S.C. § 227

4.8. Recommendations

Phone number recycling attacks can harm subscribers, yet they involve different stakeholders. As mentioned, the FCC has recently implemented an RND to help legitimate robocallers avoid placing calls to recycled numbers (§ 4.2.4). Since the database is a closed resource, it remains unclear whether this mechanism—along with access to it—can be extended to prevent any of the attacks we presented. In the meantime, carriers, websites, and subscribers can take protective measures.

4.8.1. Recommendations for carriers

1. **Warn subscribers of the risks of phone number reassignment.** Neither carrier offers any information about number recycling risks on their online interfaces. When we called T-Mobile to change the number on our postpaid account, we were briefly told to update our linked online accounts before consenting to the change. Carriers should inform subscribers that phone numbers are recycled, and provide adequate warning to them about possible threats before beginning the number change process. Specifically, carriers should ask subscribers to update any linked accounts number and to inform their peers. Carriers can also recommend subscribers keep track of any accounts tied to their new phone number upon a change (or upon account signup). That said, it is unclear whether the advice to update linked accounts is practical: according to a 2017 study, the average user has 150 online accounts [105].

2. **Publicly document number recycling policies and timelines.** Carriers should document their number recycling policies, including the ways subscribers can lose access to their numbers as well as a timeline for regaining access to them. Carriers stand to benefit from informing subscribers, as speculation on forums

varies widely. Subscribers should not be left to guess the amount of time they have to update their peers, online accounts, and bank accounts. T-Mobile and Verizon should also clearly document their policy in CSR playbooks to ensure correct and consistent responses.

3. **Place limits on phone number inquiries online.** On postpaid interfaces, Verizon already has safeguards and T-Mobile does not even support changing numbers online (Table 4.4). However, the number pool is shared between postpaid and prepaid, rendering all subscribers vulnerable to attacks. Carriers should not allow for unlimited queries at their prepaid interfaces. They can also consider restricting subscribers from viewing full numbers online, and instead direct subscribers to contacting customer service if they wish to do so.

4. **Place limits on phone number changes online.** In addition to limiting queries, carriers should limit the amount of times subscribers can request a number change. Verizon already places limits on number changes for both prepaid and postpaid, yet T-Mobile allows for unlimited number changes at its prepaid service. Without restrictions, an attacker can carry out large-scale account hijackings with a single account by constantly switching numbers (§ 4.3). Limiting number changes would essentially reduce the number of hijacked accounts an attacker can amass and sell on the dark web, hence reducing the profitability of the attack.

5. **Offer number parking for inactive subscribers.** If a subscriber knows that they will not require phone service for an extended period of time (e.g., a college student studying abroad), they should be given an opportunity to keep their number. There already are third-party services in which subscribers can store their phone number on a low-cost monthly subscription; transferring to the service cancels and removes the need to pay for their more expensive

carrier plan [106]. This is different from a voluntary / vacation suspension, which does not cancel the mobile plan and is capped at 90 days by the FCC. T-Mobile and Verizon already offer voluntary temporary service suspensions for their postpaid subscribers only.

4.8.2. Recommendations for websites

While carriers can raise awareness and provide clarification about phone number recycling, subscribers with accounts on websites relying on SMS 2FA continue to be at risk. In a study looking at SIM swaps, Lee et al. examined 2FA and recovery settings at over 140 websites, and discovered 83 sites had defaulted to SMS 2FA, which could be defeated with a phone number hijacking like a SIM swap [1]. Worse, 17 websites were *doubly insecure* (4 of which we analyzed in § 4.4); an attacker could hijack SMS 2FA-enabled accounts without knowing the passwords.

Websites need to recognize the security ramifications of their default and allowed configurations, which put accounts at risk of takeover. Our recommendations for websites are identical to that from the SIM swaps study:

1. *Doubly insecure* websites need to prevent simultaneous use of SMS for account recovery and 2FA
2. Implement at least one secure 2FA option
3. Eliminate / discourage SMS 2FA

Websites can explore more effective 2FA and recovery reminders through usable security research. One such reminder design can explicitly ask users to remove inaccessible factors—such as previous phone numbers—when reviewing 2FA options. To that end, websites should also provide support to users who no longer have phone service at all, and offer alternate forms of identity proof.

Ultimately, number recycling attacks should give further reason for websites to move away from using phone-based authentication, since they have no reasonable way of determining changes in number ownership.

4.8.3. Recommendations for subscribers

Earlier, we highlighted that subscribers may choose to keep their phone number when switching providers (§ 4.2). Number portability is regulated by the FCC and mandates that carriers allow active subscribers to switch to a competitor while retaining their original number for little to no cost (47 C.F.R. § 52.35). The transfer procedure is called *porting*. Portability facilitates seamless transition between carriers.

Porting has an added—and largely unrealized—use case: preventing reassignment of a number that a subscriber no longer wants to use. We refer to porting for this purpose as *parking*.

We recommend subscribers park their current phone numbers when disconnecting their lines. Subscribers can park their number at a dedicated parking service (e.g., NumberBarn offers low-cost monthly number parking), a mobile virtual network operator (which usually offers plans cheaper than those of major carriers), or to a VoIP provider like Google Voice (which charges a one-time fee to port in a phone number, which then never expires). This includes subscribers looking to change their number, and those who need to temporarily disconnect their lines beyond the 90-day suspension offered by some carriers (e.g., a worker contracted overseas).

Number parking mitigates several number recycling threats:

1. Subscribers now have more time to update their SMS 2FA settings.
2. Temporarily-disconnected subscribers can prevent accidental number losses from aging period confusion.

3. IPV survivors can prevent their old number from being available for reassignment for some period of time, in order to prevent abusers from taking over the old number (**Targeted takeover**).

When the subscriber is ready to release her old number, she can cancel her parking subscription. The parked number will be returned to the original carrier for recycling. Returning subscribers can resume usage by "unparking"—porting out their parked number—to their original or new carrier.

While effective, parking may not always be feasible. Number portability only allows active subscribers to move their current phone numbers; those who have already given up their number—for reasons we listed in § 4.2.2—will generally be unable to get their number back to park.

Table 4.6.: Measures that subscribers can take against the eight number recycling attacks.

Attack	Mitigating step(s)
PII indexing	Avoid unnecessarily sharing PII; opt out of people search databases [107]
Account hijackings via recovery	Avoid SMS 2FA/recovery if secure options are available; avoid *doubly insecure* setups; remove previous numbers from account settings
Account hijackings w/o password reset	Avoid password reuse; avoid SMS recovery; remove previous numbers from account settings
Targeted takeover	Park old number indefinitely; file criminal complaint for cyberstalking
Phishing	Ignore and report phishing messages, avoid clicking on links; call carrier to verify
Persuasive takeover	Ignore and report phishing messages
Spam	Report spam texts to carrier and to FCC; enable spam blocker
Denial-of-service	Ignore ransom requests to "free-up" recycled number; contact websites to manually prove ownership of number

In Table 4.6, we list steps subscribers can take to combat the threats from the eight number recycling attacks we introduced in § 4.3. For attacks affecting previous owners (**PII indexing**, **Account hijackings via recovery**, **Account hijackings without password reset**, and **Targeted takeover**), these steps should be taken with our primary recommendation to park the number (if feasible).

These mitigating steps require subscribers to be proactive. Moreover, not all steps guarantee complete protection, and some may be hindered by external factors. For instance, a website might not allow a subscriber to remove their previous recovery phone number without providing a new number—the subscriber might not have an active number. Furthermore, even if a subscriber opts out of people search databases, their PII remains publicly available on websites from which it originates. While subscribers can certainly reduce the risks of number recycling attacks with these measures, these threats remain feasible so long as phone numbers are recycled.

4.9. Summary

As a regulated industry practice, phone number recycling is unlikely to cease. We highlighted eight different security and privacy threats that are perpetuated by number recycling, and empirically showed the seriousness of three of those attacks. Although we successfully advocated for the two carriers we studied to clarify their number recycling policies for subscribers, more work can be done by all stakeholders to illuminate and mitigate the issues. In particular, online services should no longer equate a correctly-entered SMS passcode with successful user authentication.

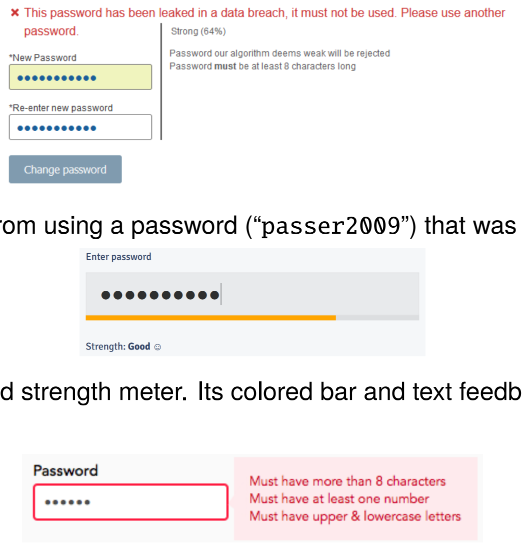
5

Password policies of most top websites fail to conform to best practices

5.1. Introduction

Passwords remain the most common means of authentication on the web, despite their shortcomings. According to industry estimates, close to half of data breaches involved authentication failures [117, 118]. As such, the need to use strong passwords remains unchanged [119]. To encourage this, websites mainly use three types of interventions during password creation: blocklists, password composition rules / policies (PCPs), and strength meters (Fig. 5.1). All three interventions have been extensively researched in the information security community.

Prior research has generally concluded that blocklists and strength meters—when configured correctly—lead users to create stronger passwords without significantly burdening them [109, 112, 120]. However, PCPs that require specific character-classes (i.e., lowercase, uppercase, digits, and symbols) are not recommended. That's because users fulfill requirements in predictable ways like capitalizing the first letter or placing a "!" at the end, negating the putative security benefits [121–123]. Additionally, character-class PCPs have consistently received poor usability ratings; in those same studies, users needed more attempts to create a compliant password and had difficulty recalling the password. Instead, websites should set only a



× This password has been leaked in a data breach, it must not be used. Please use another password.

*New Password

●●●●●●●●●●●

*Re-enter new password

●●●●●●●●●●●

Strong (64%)

Password our algorithm deems weak will be rejected
Password **must** be at least 8 characters long

(a) A website preventing us from using a password (“`passer2009`”) that was leaked in a data breach.

Enter password

●●●●●●●●●●

Strength: **Good**

(b) An example of a password strength meter. Its colored bar and text feedback changes in response to the entered password.

Password

●●●●●●

Must have more than 8 characters
Must have at least one number
Must have upper & lowercase letters

(c) A *3class8* character-class PCP, which requires passwords be at least 8 characters in length with at least 1 lowercase, 1 uppercase, and 1 number.

Figure 5.1.: Examples of the three interventions we studied: blocklists, PCPs, and password strength meters.

minimum-length requirement while complementing it with a blocklist check or minimum-strength requirement [109].

The research is clear; what is less clear is whether these best practices are actually being followed. There has been no comprehensive study to understand how online services guide their users in setting up passwords (although previous studies have looked at narrow aspects of this question [111, 124]). We aimed to fill this gap by examining password policies of 120 of the most popular English-language websites in the world. By signing up for accounts and manually testing requirements for password creation, we discovered each website’s blocklist strategy, PCP, and strength meter implementation (if any). We asked the following research questions:

1. Are websites preventing users from using the most common passwords? (§ 5.3)

Table 5.1.: We contrast our key findings with established best practices for encouraging strong passwords.

	Best practices from prior research	Our key findings
Blocklists (§ 5.3)	• Do check users' passwords against lists of leaked and easily-guessed passwords [7, 108–110]. • Do reject the password if it appears on a blocklist, prompt the user to select a different password [7, 110].	• More than half (71 / 120) of websites do not check passwords at all, allowing all 40 of the most common passwords we tested (e.g., "`12345678`", "`rockyou`"). • 19 more websites block less than half of the most common passwords we tested.
Strength meters and min-strength reqs (§ 5.4)	• Do provide real-time password strength estimates [111–113]. • Do set minimum-strength requirements by estimating guessability (the number of guesses it would take for an adversary to crack the password) [44, 47, 109, 114, 115].	• Only 23 / 120 websites used password strength meters. • Of those 23, 10 websites misuse meters as nudges toward character-class PCPs and do not incorporate any notion of guessability.
Composition policies (§ 5.5, § 5.6)	• Do not require specific character-classes; let users freely construct passwords [108, 109, 113, 116]. • NIST: Do set a minimum-length of at least 8 characters [7].	• 54 / 120 sites still use character-class PCPs. • We devised a new method to measure the security and usability of all 120 PCPs. Based on our method, we found that all PCPs performed poorly, none provided $\geq 60\%$ security and usability simultaneously.

2. Are websites using password strength meters to encourage strong passwords? (§ 5.4)

3. What PCPs are used by top websites? What are the security-usability tradeoffs of those PCPs? (§ 5.5, § 5.6)

We considered a website to be following best practices if it simultaneously satisfied the following security and usability criteria:

- **Security:**
  - Allowed 5 or fewer of the 40 most common leaked passwords and easiest-to-guess passwords (e.g., "`12345678`", "`rockyou`") we tried.

  - Required passwords be no shorter than 8 characters OR employed a password strength meter that accurately measured a password's resistance to being guessed by an adversary [113].

- **Usability:** Did not impose any character-class requirements.

We found that only 15 websites were following best practices. The remaining 105 / 120 either failed to adhere or explicitly flouted those recommendations in their policies, leaving users at risk for password compromise or frustrated from being unable to use a sufficiently strong password. We compare our key findings with the best practices for all three interventions in Table 5.1.

We further devised a method to measure the security and usability of PCPs using a large corpus of breached passwords. Past studies have typically examined a small number of different PCPs due to constraints with hiring participants, which motivated us to design a method that could scale to the large number of PCPs we examined. These studies have also systematically neglected to investigate PCPs with short minimum-length requirements, which we frequently found during our study (the following paragraph suggests a reason why previous studies may have excluded these PCPs). While we were able to analyze the PCPs of all 120 websites we visited, we note that our strategy has limitations and should be used to complement findings from previous user studies. We found that no PCP had more than 60% security and usability simultaneously. These results further call into question some of the recommendations on PCPs that have been taken at face value, without any evidence.

While there is broad consensus on best practices in the prior literature, it is sometimes unclear exactly where to draw the line. For instance, the National Institute of Standards and Technology (NIST) recommends an 8-character minimum-length requirement in its current version of *Digital Identity Guidelines*—a widely relied-upon resource by both practitioners and researchers [7]. Yet, that recommendation does not cite any research. Even though we performed a thorough literature search and failed to find any research that had investigated the usefulness of setting an 8-character minimum-length, we decided to count that recommendation as a best practice. Here, we have used our best judgment in defining what constitutes best practice, erring on the side of being lenient. While our exact number might change if we change our definition of "best practices", our qualitative finding—that most websites are not following best practices—does not change.

Our findings reveal a disconnect between industry and the research community. Passwords have been heavily researched, yet few websites have implemented password policies that reflect the lessons learned. Researchers should make sure their findings have societal impact by engaging in outreach to website operators about their password practices.

5.2. Overview of password best practices

Websites mainly have three different ways to encourage users to create more secure passwords, as outlined by NIST [7]. Here we discuss previous research on the methods and their best practices.

5.2.1. Blocklists work, but need to be carefully configured

One simple way for websites to encourage more secure passwords is to keep a list of common insecure passwords (e.g., "`123456`", "`!QAZ1qaz`") and deny users from choosing passwords from that list (Fig. 5.1a). Prior research has found that password blocklists work. Kelley et. al (2012) analyzed passwords created under blocklists of different sizes and matching strategies for several password composition policies, and found that larger blocklists with more sophisticated matching algorithms led to stronger passwords being created [108]. Shay et al. (2015) found that blocklists generally increase security without sacrificing password recall among users [113]. Habib et al. (2017) also supported using blocklists, and further recommended that websites also restrict users from submitting simple modifications to blocklisted passwords [120].

Blocklists may consist of common passwords gathered through different strategies, including commonly used passwords that have been exposed in data breaches and passwords that are likely to be guessed easily by password cracking tools. Websites may also have different approaches to checking passwords against the blocklist; for instance, some may perform exact matching while others strip out symbols before matching. While NIST recommends that websites block common passwords, it is neither prescriptive on which lists to use nor on the comparison method [7].

The National Cyber Security Centre (NCSC) provides more concrete guidance [110]. In collaboration with *Have I Been Pwned?* (HIBP)—an online service that allows users to check whether their credentials have been compromised in data breaches—the NCSC has released a list of the 100,000 most common passwords for websites to use as a blocklist (which we refer to as NCSC-HIBP-100k later on in the paper). NCSC guidance reasons that blocking the top 100,000 passwords prevents users from "making poor password choices, whilst not making it too difficult for them to choose one."

Tan et al. (2020) later investigated the security-usability tradeoffs of blocklist requirements and found that blocklists—while effective—can cause user frustration if not properly configured [109]. They recommend blocking passwords that appear in NCSC-HIBP-100k or blocking common passwords that appear in a corpus of 10 million leaked passwords, both of which we used in our experiments.

In this study, we empirically examine whether websites follow the best practices for blocklists established by prior work.

5.2.2. Min-strength requirements and strength meters are both effective and user-friendly

A newer approach to encourage strong passwords is to set minimum-strength requirements. When a user submits a candidate password, the website estimates the strength for the submission, and if it is greater than the minimum threshold, the candidate password is accepted. A strength meter that updates in real-time is often shown to nudge users as they craft their passwords (Fig. 5.1b).

To measure strength, researchers recommend and typically use adversarial guessing—the number of guesses needed to crack a password (i.e., the guess number or guessability). Previously, strength was often modeled using Shannon entropy—a function of the length and number of character-classes present in a password, or its complexity. However, complexity has since been deprecated since it is not a good proxy for guessability (see Chapter 2 for further background).

Estimating password strength is difficult, especially considering that users expect near-instantaneous feedback when setting a password. Previous research has found that among the password-strength meters in use on the web, most actually measured complexity instead of guessability, and were actually inconsistent with one another (de Carnavalet et al., 2014) [111]. There was an open-source implementations that were found to be reliable, however: `zxcvbn` outputs accurate strength estimates through 10^6 guesses, the threshold for online attacks [125].

In 2017, Ur et al. designed a data-driven strength meter that estimates password strength using a client-side neural-network created in a prior study (Melicher et al., 2016) [114]. Their meter received positive feedback from participants in the following user study, and was accurate when compared with results from password cracking tools, which were used as ground truth [112]. Tan et al. (2020) later updated the meter to enforce blocklists and minimum-strength requirements, while also making the meter freely available for others to use [109]. They concluded that minimum-strength requirements are the best way to encourage strong passwords, and recommend setting the minimum-strength threshold to at least 10^6 to prevent online guessing attacks [109]. Since their password-strength meter directly estimates guessability—as opposed to PCPs indirectly using complexity as a measure of strength—websites need only set a minimum-guesses threshold instead of character-class requirements, such as 10^6 for online attacks and 10^{14} for offline attacks. They further highlight that the meter's underlying neural network can be "easily retrained to reflect changing patterns in passwords over time" and that its configurable integration with blocklists can penalize common passwords.

5.2.3. Character-class PCPs should not be used

To enforce the use of strong passwords, websites have employed password composition policies (PCPs). PCPs are rules which users must follow in creating their passwords. These rules most often include a minimum password length requirement along with character-class requirements (Fig. 5.1c). PCPs fall into two categories: ones with character-class requirements (which we'll refer to as "character-class PCPs" throughout the paper) and ones without (PCPs that only have a minimum length requirement, which we'll refer to as "minimum-length PCPs").

As a vestige of when password strength was modeled by Shannon entropy, character-class PCPs force users to create complex passwords, and prior research has found that PCPs requiring more character classes generally produced stronger passwords overall (Komanduri et al., 2011), (Kelley et al., 2012) [108, 116]. However, character-class PCPs have poor usability. Users have found it difficult to comply with the complex rules and to remember the password they have created, and several studies have even found that some users may respond predictably to comply with character-class requirements when creating passwords, which negates the benefits of adding complexity (Shay et al., 2010), (Weir et al., 2010), (Ur et al., 2015) [45, 122, 123].

As studies that have found that increasing minimum-length requirements while reducing character-class requirements can lead to strong passwords without decreased memorability, NIST has also updated its guidance to recommend websites remove character-class requirements (Kelley et al., 2012), (Shay et al., 2014) [108, 121]. It further recommended that websites require passwords be at least 8 characters long [7]. Tan et al. (2020) recently found that character-class PCPs do not make it harder for attackers using modern-day cracking tools to guess with passwords, since users now tend to incorporate multiple character-classes of their own accord. Still, they recommend against using character-class PCPs because users still find them annoying and some users will still fulfill requirements in predictable ways [109].

Even with the updated recommendations, character-class PCPs may remain ubiquitous, though they are largely unmeasured; the only previous study that explored PCPs on the web was from 2010 [124]. In our study, we measured the state of security and usability of PCPs present on the web by extracting them from websites we visited.

5.3. Study 1: password blocking

We measured whether popular websites prevent users from choosing the most common insecure passwords and found most of them insufficiently block users' choices. We selected common passwords to test based on two different strategies:

blocking the 100,000 most frequently-used passwords found in password breaches (NCSC-HIBP-100k) and blocking passwords guessed early on by state-of-the-art cracking tools.

5.3.1. Method

5.3.1.1. We tested 120 of the top websites

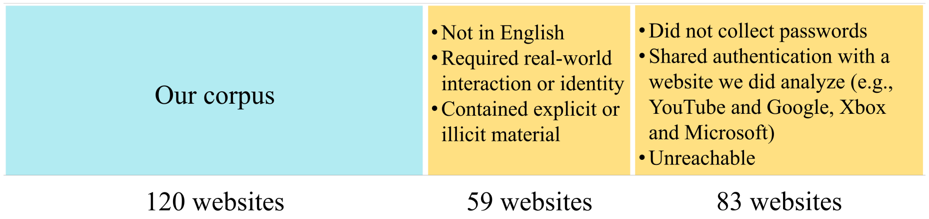

Our corpus
• Not in English
• Required real-world interaction or identity
• Contained explicit or illicit material
• Did not collect passwords
• Shared authentication with a website we did analyze (e.g., YouTube and Google, Xbox and Microsoft)
• Unreachable
120 websites
59 websites
83 websites

Figure 5.2.: A breakdown of the 262 websites we attempted to study. We skipped 59 websites that did not fit our selection criteria. 83 websites either could not be analyzed or were already represented among our corpus of 120 websites.

In this study, we are concerned with password policies at the most popular English-language websites so our findings could be verified by all co-authors (who are all fluent in English). We focused on popular websites because previous research has shown that they generally have better security policies [124], which means that our results can be seen as an underestimate of conformity with best practices. Further, we wanted to hold these specific websites to account because they affect more users. Using an actively maintained ranked list provided by other researchers [126],[1] we tested the top 120 websites that were accessible to us. We skipped some websites for the reasons shown in Fig. 5.2; we reached our total of 120 websites after trying the top 262 listed entries.

[1] Available at `https://tranco-list.eu/list/VJ5N`. Generated on 29 July 2021.

Before the tests, we extracted the PCP on each website and encoded them in a regular expression (detailed in § 5.5.1). This allowed us to select PCP-compliant passwords for testing.

5.3.1.2. Testing common passwords leaked in breaches

We sampled 20 passwords from the NCSC-HIBP-100k list, which was ordered from most common to least common. We started by removing passwords that did not fit the website's PCP (with the aforementioned regular expressions) and sampling candidates to test at each website. In order to evenly represent the most frequently-leaked passwords along with the long tail of rest of the passwords in the list, we used a stratified sample based on powers-of-10 (1-10, 11-100, 101-1,000, 1,001-10,000, and 10,001-100,000). We randomly sampled candidates weighted by their position on the list ($\frac{1}{position}$), which gave us—in expectation—4 passwords in each stratum. In order to ensure fair comparisons, websites with identical PCPs were tested with the same 20 passwords (e.g., all websites with a *1class8* PCP were tested with "`babygirl23`", "`lifeisgood`", etc.) We refer to these tested passwords as *leaked* passwords hereinafter.

Using the accounts we had set up, we attempted to change our password to each of the *leaked* passwords. If the change was successful, we logged out and logged back in with the new password to confirm, then noted that the password was accepted.

5.3.1.3. Testing common easy-to-guess passwords

In addition to restricting *leaked* passwords, websites should discourage users from selecting common passwords that are easily guessed (e.g., block "`Blink182`", which can be guessed in ~ 9 tries, or "`Hello123`", which can be guessed in ~ 316 tries). Here we tested the first 20 passwords that were guessed by state-of-the-art cracking tools at each website. We refer to these tested passwords as *easiest-guessed* passwords hereinafter.

We used Password Guessability Service (PGS)—offered by the Passwords Research Team at Carnegie Mellon University—to find these passwords to test [24]. PGS simulates a real attacker guessing passwords; it leverages multiple (5, at the time of our study) cracking tools to arrive at the user-provided plaintext password, returning the guess number (i.e., the number of guesses needed to find the password) as the password's strength rating. PGS also offers the `min_auto` configuration, which returns the minimum guess number for each password across all 5 tools. Previous research has found that the `min_auto` approach provides a conservative estimate for the performance of an unconstrained professional attacker [24]. Therefore, we referred to the `min_auto` guess number for all of the passwords in this study.

Since PGS requires its users to provide passwords in plaintext in order to receive results, we selected passwords to use from the Xato 10-million password dataset [127]. To the best of our knowledge, the dataset—which we will refer to as the Xato passwords hereinafter—represents the largest and most recent corpus of real-world passwords accessible to academic researchers, and has been widely used in previous work [109, 112, 120, 125]. We did search for newer password dumps to complement the Xato passwords, but found they were either available only on the dark web or offered in hashes rather than plaintext (to prevent large-scale cracking) [128, 129].

With all of the Xato passwords rated, we used the 20 passwords with the lowest guess numbers as our *easiest-guessed* passwords, and tested whether websites allowed them to be used. As with our testing of *leaked* passwords, we only selected passwords that fit the website's PCP. We excluded passwords that were already in the *leaked* passwords, and selected the password with the next-lowest guess number instead.

Here, we also tested the same 20 passwords across websites with identical PCPs to ensure fairness (e.g., all websites with a *DigSym6* PCP—6+ characters with 1 digit or symbol—were tested with "`jordan23`", "`jessica1`", etc.). Every selected password could be guessed within $10^{4.9}$ guesses, well within the threshold of online guessing attacks.

5.3.2. Results

1. **Most websites do not block *leaked* or *easiest-guessed passwords* at all.** 71 / 120 websites accepted all 40 passwords we tested. By allowing both *leaked* and *easiest-guessed* passwords, these websites put their users at risk of password compromise and subsequent account hijackings. Additionally, accounts at other websites may be at risk for compromise too; users often practice poor security hygiene by reusing their passwords across the web, so this misconception that their password was not blocked and therefore suitable can have widespread insecurity.

   These 71 websites span different industries, including e-commerce (Amazon), social media (TikTok), entertainment (Netflix), and news (Wall Street Journal). Amazon, for instance, allowed us to change our password to "`123456`", the most common password on the web. TikTok—despite requiring users to choose a *3class8* password—allowed us to use "`p@ssw0rd`" (guessed by PGS in 7 tries, the fourth most common *3class8* password) on our account.

2. **Additionally, several websites had insufficient blocking.** In addition to the 71 websites which accepted all 40 passwords, 19 sites accepted more than half of the *leaked* or *easiest-guessed passwords* tested. In some of these cases, this was likely due to insufficient blocklists. For example, IBM seemed to only block choices containing the word "password", which only blocked 1 ("`Password1`")

Table 5.2.: We found 10 websites that seemed to be blocking passwords based on a shorter common passwords list, and found 7 websites that seemed to be blocking passwords that did not meet a minimum-strength requirement.

	Fraction of accepted *leaked* passwords by stratum				
	1-10	11-100	101-1,000	1,001-10,000	10,001-100,000
bit.ly	0/2	1/3	1/4	1/6	3/5
chase.com	0/1	1/3	1/4	5/7	4/5
espn.com	0/2	1/3	2/4	6/6	5/5
facebook.com	0/2	1/3	0/4	3/6	4/5
instagram.com	0/2	1/3	0/4	3/6	4/5
slack.com	0/2	1/3	1/4	6/6	5/5
spotify.com	0/2	2/3	1/4	4/6	5/5
surveymonkey.com	0/2	2/3	1/4	4/6	4/5
tripadvisor.com	0/2	0/3	0/4	6/6	5/5
yelp.com	0/2	1/3	4/4	5/6	4/5

(a) Looking at accepted *leaked* passwords by stratum, we hypothesized 10 websites were using shorter verisons of the NCSC-HIBP-100k list.

	Hypothesized minimum-strength threshold
indeed.com	10^3
linkedin.com	$10^{2.3}$
microsoft.com	$10^{2.4}$
roblox.com	10^1
reddit.com	10^1
twitter.com	$10^{3.4}$
wetransfer.com	$10^{1.5}$

(b) Minimum-strength thresholds we hypothesized were being used at 7 websites. For reference, the threat of online guessing attacks ends at 10^6 guesses.

of the 40 passwords we tested on its site. Samsung only blocked number sequences (e.g., "123"), and Salesforce only blocked "`password`". While this may prevent users from using the most guessable passwords, the majority of the most common passwords still get accepted.

3. **10 websites seemed to be using a shorter *leaked* passwords blocklist.** We found 10 websites that blocked most of the tested *leaked* passwords from the

higher-rank strata (e.g., 1-10, 11-100) but then allowed a majority of leaked passwords from the lower-rank strata (e.g., 10001-100000). This could indicate that these websites are using a truncated version of the NCSC-HIBP-100k list to check passwords, sacrificing security for usability. Spotify, for instance, blocked all *leaked* passwords up to the 101-1000 stratum but allowed all passwords beyond that point, which suggests that it only checks for the top 1000 *leaked* passwords. Our finding here is tentative, however, since we assumed websites were using the NCSC-HIBP-100k list. Table 5.2a shows the breakdown by stratum for the 10 websites.

4. **7 websites blocked *easiest-guessed* passwords, but not *leaked* passwords.** 7 websites disproportionately accepted more *leaked* passwords than *easiest-guessed* ones (Microsoft: 14 *leaked* accepted / 0 *easiest-guessed* accepted, LinkedIn: 14 / 0, WeTransfer: 19 / 4, Roblox: 18 / 6, Reddit: 16 / 4, Twitter: 12 / 2, and Indeed: 9 / 1).

   These 7 websites might have been using a minimum-strength requirement instead, since passwords they accepted generally had higher guess numbers than the passwords they rejected. If true, none of the websites set their minimum-strength requirement to prevent the threat of online guessing attacks (10^6 guesses). For example, Microsoft accepted one *leaked* password cracked with 251 guesses, and WeTransfer allowed one *leaked* password cracked with 32 guesses. Table 5.2b shows the minimum-strength cutoffs we found through our testing.

5. **Few websites prevented us from setting *leaked* and *easiest-guessed* passwords.** Only 15 websites—including Google, Adobe, Twitch, GitHub, and Grammarly—blocked all 40 passwords we tried. 7 more websites—including

Apple, Canva, and VK—performed moderately well, allowing 5 or fewer tested passwords.

6. **Websites that allowed *leaked* and *easiest-guessed* passwords hold sensitive user information.** 38 of the 71 websites that allowed all 40 passwords store user payment information such credit card or banking details, including Amazon, Netflix, Flickr, GoDaddy, and Squarespace. 64 / 71 websites store PII about users, including Line, Intuit, Zoom, IMDB, and MySpace. For each of the websites we analyzed, we checked whether it stored sensitive information using the test account we created earlier. Without additional measures like 2FA (which may not be available at every website), these users are at risk for identity theft or payment fraud.

§ C.5 contains a table that shows the number of accepted *leaked* and *easiest-guessed* passwords for all 120 websites.

5.4. Study 2: strength meters

In 2014, de Carnavalet et al. investigated 11 password strength meters that were used in practice, and found most were estimating password complexity instead of guessability [111]. We wanted to know if there had been any changes; are meters at top sites now estimating guessability when a user chooses a password? To answer this, we reverse-engineered their patterns by testing different passwords.

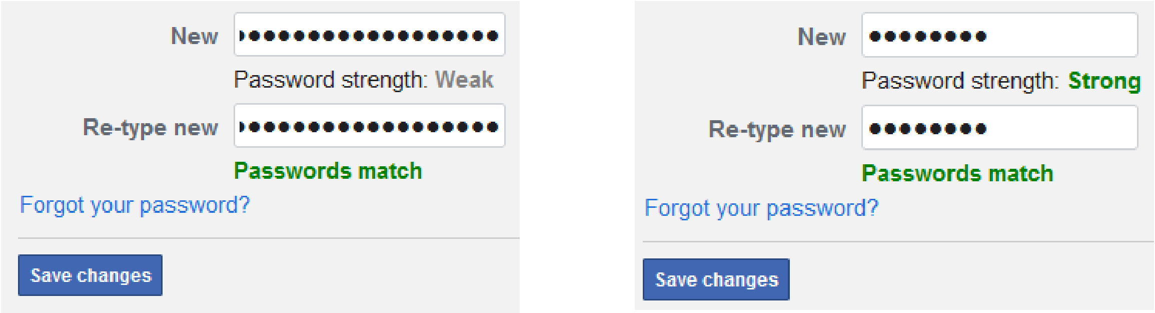
5.4.1. Method

We considered all form elements on the password update page that updated in real-time to give feedback about the strength of the input on the password field. We then ran two tests on each of the strength meters to learn its patterns. First we investigated whether the meter was consistent in discouraging insecure choices; we tested the 100 *easiest-guessed* passwords from Xato that fit the website's PCP and noted the feedback received on each password. Next we tried to reverse-engineer the mechanics of the meter through boundary testing. We tested passwords with different lengths and number of character-classes, as well as passwords that were not compliant with the website's PCP. We selected passwords from Xato and also used passwords generated from password managers—Lastpass and 1Password—and noted movement patterns along the strength meter.

5.4.2. Results: most websites are not using strength meters to measure guessability

Table 5.3.: Of the 23 websites that used password strength meters, 10 used those meters to encourage more complex passwords. 6 websites with minimum-length PCPs were actually using their meters as proxies for character-class PCPs.

Password strength meters at websites with:	Our finding(s)	Implications on users	Prevalence
Minimum-length PCPs	• **Encourages complex passwords over passwords that are harder-to-guess.** Rates *easiest-guessed* passwords that have more character-classes higher than passwords with high guess numbers but fewer character-classes. • Discourages users' choices by nudging them toward fulfilling character-class PCPs.	• Password strength feedback does not reflect password guessability. • Possible usability issues similar to when fulfilling character-class PCPs.	6 / 18
Character-class PCPs	• **Encourages more complexity than required.** Meter tops out only if passwords include more character-classes than required by the PCP.	• Password strength feedback does not reflect password guessability. • Usability issues may arise when a candidate password meets all stated requirements but does not fill the meter.	4 / 5



Password strength: Weak

Passwords match

Password strength: **Strong**

Passwords match

(a) “`bkmmafwexucnvnsgppdk`” (1 class, randomly generated) rated as Weak (1/3).

(b) “`Passw0rd`” (3 classes) rated as Strong (3/3).

Figure 5.3.: Despite having a *1class6* PCP, Facebook’s password strength meter is driven by adding more character-classes, and not password strength.

1. **Password strength meters are not widely used.** We found only 23 websites using password strength meters of any sort. Despite previous research touting the added security and usability benefits of using strength meters and robust open-source implementations like `zxcvbn`, most websites have not updated their password change procedures.

   Regarding recommended strength meters, we found only 2 websites using `zxcvbn`; Dropbox (the organization behind the meter) and CPanel. The rest of the websites were using black box implementations that may not have been rigorously tested by the research community.

2. **10 / 23 websites misuse strength meters to measure complexity instead.** Rather than measuring password guessability, we found meters were actually being used as proxies for character-class PCPs. We break down our results by PCP here and in Table 5.3:

   6 / 18 websites with minimum-length PCPs use strength meters as character-class PCP nudges. Their strength meters would only increase if a password had more character-classes than the one entered prior, and not if it had a higher guess number. Despite Facebook’s *1class6* PCP, its 3-point strength meter—shown in Fig. 5.3—considered all-lowercase passwords weak; “`zcdplgbtqldecfrzdqrw`” (randomly generated) was considered Weak (1/3) while “`Password1`” was considered Strong (3/3). The strength meter at Yelp (*1class6* PCP) unconditionally considered 16-character passwords strong, while requiring shorter passwords contain all four character-classes to be considered as such; “`123456789123456789`” (guess number ∼ 631) was considered Great (4/4) while “`WzNGVE5uuWHd`” (randomly generated) was considered only Good (3/4). Since these meters measured complexity instead of estimating guessability, their readings were not reflective of how difficult it would be for an adversary to crack the password. Furthermore, users are nudged into creating complex passwords at these sites. The other 12 websites with minimum-length PCPs—including Google, Yahoo, and Twitch—had strength meters that more closely corresponded with password guessability; they rated all 100 passwords we tested as weak (<50% on their respective meters), and we did not find any insecure patterns when testing passwords with different character-classes and length.

4 / 5 websites with character-class PCPs use their meters to encourage further complexity. They reserved the highest ratings on their meters for passwords that went beyond the required character-classes. Apple’s strength meter, for instance, would only reach 100% if the password was 16-characters long and contained a symbol, despite its corresponding *3class8* PCP not requiring symbols. Aliexpress’s 3-point meter only topped out if all 4 character-classes were included, despite a *2class6* PCP; “`jmDy&!py$Df&^tw*iBYy`” (randomly generated 3-class) was rated Middle (2/3), yet “`Abc123!@#`” (guess number ∼ 53) evaluated to High (3/3). Since users may already be led by these sites to believe that compliance with character-class requirements would automatically yield strong passwords, they may find it frustrating when their password does not top up the strength meter. We only found one website—ScienceDirect—which did not encourage further complexity, only because its meter already filled up completely upon PCP-compliance.

3. **12 / 23 websites were inconsistent between meter feedback and password acceptance.** We then raised the question: is the feedback from the meter on the user-side consistent with the ultimate decision by the website to accept or reject a user's chosen password on the server-side? Here, we used findings from our password blocking analysis—in which we had selected the 20 *easiest-guessed* Xato passwords that were compliant with a website's PCP and tested whether the website would accept them (§ 5.3)—and compared them with feedback given by the website's password strength meter. We now focused on feedback given by the website's strength meter right before submitting each password to the server. For each password, if feedback from the strength meter was negative (i.e., <50% of the scale), we coded user-side feedback as "unacceptable," otherwise, we coded the feedback as "acceptable."[2]

   12 / 23 websites had varying levels of inconsistency. 5 websites rated all 20 passwords as "unacceptable," yet the server allowed all of them to be used; these websites rely solely on their strength meters, and do not perform additional checks before updating passwords. At CPanel, all 20 tested passwords were "unacceptable" (we found it was using `zxcvbn`), yet the server only rejected 13, which had all-letters or all-repeating-digits (e.g., rejected "`66666666`" but accepted "`12345`").

[2] Fortunately, we did not have to deal with any ambiguity between scale readings and labels on the meters we saw; all points below 50% had negative feedback, and all points 50% and above had neutral or positive feedback.

> Only 11 / 23 websites were consistent between their strength meter feedback on the user-side and acceptance on the server-side. 8 of those sites—including W3C, Tumblr, and TechCrunch—rated all 20 passwords as "unacceptable," and all 20 were ultimately rejected. The other 3 sites were consistent in the opposite manner; they rated all 20 passwords "acceptable," and all 20 were ultimately accepted. Overall, these inconsistencies can lead to insecurity stemming from users unknowingly setting easiest-guessed passwords, as well as frustration when a user is told a password is good enough but is rejected.

Our key finding is that despite the usefulness of password strength meters being established in the research literature, adoption has remained low, and 10 / 23 of the sites that have them—6 of which have minimum-length PCPs—actually misuse them as proxies for character-class PCPs.

5.5. Study 3: composition policies

5.5.1. Method

We reverse-engineered the PCPs of the 120 websites we visited and analyzed them.

5.5.1.1. We extracted the PCPs on 120 of the top sites

Using the aforementioned Tranco list (§ 5.3.1), we visited the top 120 websites that were accessible to us. At each website, we created an account and subsequently navigated to the password change page to reverse-engineer the website's PCP. We chose to use the password change page over the account creation page in order to avoid the need to repeatedly create new accounts and enter sign-up information (e.g., usernames, email addresses, names).

We noted the static creation rules that loaded on the form, then extracted dynamic rules by varying the password input with sample strings we had prepared in advance. We varied our sample strings to include strings with 1 class only (all lowercase letters, all digits, etc.) and strings with multiple classes (uppercase, lowercase, digits, and symbols). For symbols (i.e., special characters), we limited our permutations to the 33 ASCII characters that could be typed on a standard U.S. keyboard (shown below; note presence of the space character):

```
 !"#$%&’()*+,-./:;<=>?@[\]^_‘{|}~
```

We prioritized completeness in our method. For each PCP, we input and submitted non-compliant sample strings to make sure the website was enforcing its shown PCP. We also tested classes and characters that were not explicitly stated (e.g., for a hypothetical "`include at least one number`" character-class PCP we tested a sample string with only numbers to make sure there was no letter requirement, for a vague "`include a special character such as !#@( `" PCP we tested all 33 symbols and occasionally found websites that 1) counted other symbols towards the requirement, 2) allowed but did not count other symbols towards the requirement, or 3) disallowed other symbols entirely. The entire extraction task was done by hand and recorded in a spreadsheet (see § C.2 for a discussion of our attempts at an automated pipeline) by one of the co-authors and verified for correctness by a second co-author. After verification, we encoded the raw text of each website's PCP into a regular expression (which was also verified by at least two co-authors). The regular expressions were later used for other analyses in our study. We ended up with 73 different regular expressions (hence, 73 distinct PCPs among the 120 websites).

Table 5.4.: Character-class requirements on the 54 websites with character-class PCPs. Nearly all require that passwords include a digit.

Character-class requirement	Websites (N=54)
Lowercase letters	31 (57.4%)
Uppercase letters	30 (55.6%)
Letters (case-insensitive)	19 (35.2%)
Digits	53 (98.1%)
Symbols (special characters)	37 (68.5%)

5.5.2. Results: character-class PCPs are still widely used

Our findings are as follows:

1. **Character-class PCPs are still widely used.** 54 websites (45%) still require users to include specific character classes in their password, despite recommendations against these requirements. As found in previous studies, character-class PCPs impose a huge usability cost for a minimal security benefit [121, 123]. Table 5.4 shows the breakdown of the required character-classes. Almost all character-class PCPs require a digit, and symbols were the second-most popular requirement.

2. **Websites with character-class PCPs are more likely to allow the most common insecure passwords.** Cross-referencing our findings from § 5.3.2, we found that 38 of the 54 websites (70.4%) with character-class PCPs accepted all 40 of the *leaked* and *easiest-guessed* passwords we tried, compared with 33 of the remaining 66 websites (50%) with minimum-length PCPs. This may suggest that websites believe that complexity requirements are sufficient in getting users to create strong passwords, so they do not need to check passwords on a case-by-case basis.

3. **The most common minimum-length requirement is now 8 characters.** In 2010, Bonneau and Preibusch found that 52% of websites studied were using

6-character minimum length—followed by 4 characters (14%) and 5 characters (10%)—and that less than 5% of websites studied had an 8-character minimum length [124]. In our results over a decade later, we found that 66 / 120 websites studied (55%) have an 8-character minimum length, followed by 6 characters (35 / 120) and 5 characters (7 / 120). Perhaps this is a result of updated guidance from NIST in 2017, which now recommends an 8-character minimum length for passwords, up from its previous recommendation of 6 characters [7, 46].

4. **9 websites had inconsistencies between the PCP and text shown.** 1 website mentioned only a minimum-length requirement, but we were unable to save our password unless it contained a digit or a symbol. 2 other websites similarly failed to mention an additional character-class requirement in their text, which we uncovered through our testing. On the flip side, 4 websites did not enforce all of the character-class requirements mentioned. For example, Canva seemed to require us to include "`a mix of letters, numbers & symbols`" in our password, but we found that there were actually no character-class requirements. 1 website mentioned that whitespace characters were not allowed, but still accepted our password containing it. Lastly, 1 website with a *2class8* PCP had no text at all. We were only able to reverse-engineer its PCP after opening up development tools on our browser to view the server responses and making multiple attempts with different character-class combinations. Overall, these inconsistencies can lead to a confusing user experience.

Our key finding is that character-class PCPs are still being used on 45% of popular websites, burdening users while providing minimal security benefit. Even with the research against these complexity requirements, websites continue to force users to include extra characters like digits or symbols in their passwords, which some users may respond to in predictable ways. Furthermore, over 70% sites that continue to use character-class PCPs do not have any other password checks in place, allowing *leaked* and *easiest-guessed* passwords to be used.

We document several additional findings in § C.3.

5.6. Study 4: PCP security and usability

In previous studies on PCPs, researchers typically conducted user studies by recruiting thousands of participants online (e.g., on Amazon Mechanical Turk) to perform password creation tasks on a testbed website. They would then analyze the passwords created for each PCP, such as measuring the complexity (entropy) of passwords created under each condition, the fraction of passwords guessed at a given guessing threshold, number of failed attempts, user sentiment, time taken to create a compliant password, and password recall rate [108, 116, 120]. These studies have influenced changes in password best practices over the past decade, particularly with the recommendation against character-class PCPs [7].

While it would certainly be useful to perform the same kind of password creation study for real-world PCPs, this was not feasible due to the large number of experimental conditions. As mentioned in § 5.5, we uncovered 73 unique PCPs among the 120 studied. For reference, in a previous user study, the authors recruited 5,099 participants who were assigned to 15 different PCPs; in order for us to replicate that power, we would have to recruit nearly 5 times as many participants [109]. These previous studies have also recognized the same limitations, and have kept the number of PCPs tested relatively small [108, 120, 121].

We therefore devised a different approach to measure the security and usability of PCPs studied. As we will show, our method has both advantages and limitations.

Therefore our findings are tentative, and are ultimately intended to complement the findings from previous studies by providing insight into PCPs in practice.

5.6.1. Method

The fundamental insight of our method is to consider a PCP as a binary classifier, whose goal is to reject weak passwords and accept strong, hard-to-guess passwords. Here, we defined a password as weak if PGS could guess it in an online attack (within 10^6 guesses), and strong otherwise.

5.6.1.1. We assumed a corpus of passwords created without constraint

Users have different ways of generating passwords that are not influenced by a website's PCP [130]. Some examples include:

- Using a fixed password for all websites (password reuse)
- Using a password manager to automatically generate passwords
- Using a fixed heuristic (e.g., dictionary word + digit)

For our analysis we needed a sample of these "unconstrained passwords" to make unbiased comparisons of security and usability across PCPs. Our sample used here is the Xato 10-million passwords set (56% / 44% split between strong and weak passwords) [127]. Even though it did not meet our requirement of passwords generated without constraint—because users were already subject to the PCPs of the breached websites in this set—we still used it for analysis. We discuss the implications of using the Xato passwords later on.

5.6.1.2. We used sensitivity as a proxy for security

We assumed that whenever a user sets a new password at each of the 120 websites we studied, they initially generate one using an unconstrained strategy. If allowed by the PCP, the user will then confirm and set the password. Any PCP will allow some fraction of weak passwords through, however, which is why we measured the sensitivity of the PCP. We consider sensitivity—the percentage of weak passwords rejected—as a proxy for security. A website that simply allows any password to be used (i.e., no PCP), for example, would have 0% sensitivity.

One advantage of using sensitivity is that it is unaffected by outliers. Some generated passwords may have extremely high guess numbers and thus skew the average strength of passwords accepted by a PCP, for example. Since we used PGS to obtain guess numbers for the Xato passwords, we also benefited from accurate password strength ratings, as opposed to using entropy [45]. Our method to measure security has one disadvantage, however. Unlike in an intervention study, we don't know how users will react to any of the 120 PCPs, such as whether users go on to create strong passwords [116].

5.6.1.3. We used specificity as a proxy for usability

Some users—given their strategy for unconstrained password generation—will be frustrated by the PCP and be forced to pick a different strategy and password. While the usability cost would be justified if their password was actually weak, it would not be justified if it was strong. In the case of a password manager being incompatible with a PCP, they may be forced to pick a password manually, making it both weaker and less memorable (see § 5.2). Here, we used specificity to measure this usability cost. We used this measure as a proxy for usability of the PCP.

Specificity is an objective measure that complements other usability measures, like recall, user dropout, and time taken to enter a password. The main disadvantage to using specificity, however, is our inability to gauge user sentiment. That is, users may not necessarily feel frustrated by the PCP if their unconstrained password generation strategy is unsuccessful, especially if they have repeatedly encountered the same kind of PCP and have (predictable) adaptation strategies, or if their password manager accommodates them [131].

5.6.1.4. Limitations of using Xato passwords

Finally, we revisit the assumption about having a corpus of unconstrained passwords. Unfortunately, the Xato passwords set does not satisfy this requirement. While it is incredibly diverse—with weighted samples of over 1,000 password dumps collected over at least 5 years—most of the passwords were probably created by users reacting to some PCP [132]. One advantage to using the dataset, however, is that it doesn't contain passwords that required cracking [132]. This means that there is no bias towards weaker passwords.

Ultimately, using the Xato passwords in our security / usability evaluation means that we will overestimate usability (e.g., the segment created under the same PCP as the one being tested will have 100% usability) and underestimate security (e.g., the segment created under the same PCP as the one being tested will have 0% security). We reiterate that our findings in this section should be regarded as tentative; yet the strong limitations of PCPs that they reveal call into question the usefulness of PCPs and call for further research using different corpora and/or methods. For instance, a future user study could ask users to create passwords under no constraint (i.e., "`include at least 1 character`") and make that password set available for other researchers to use.

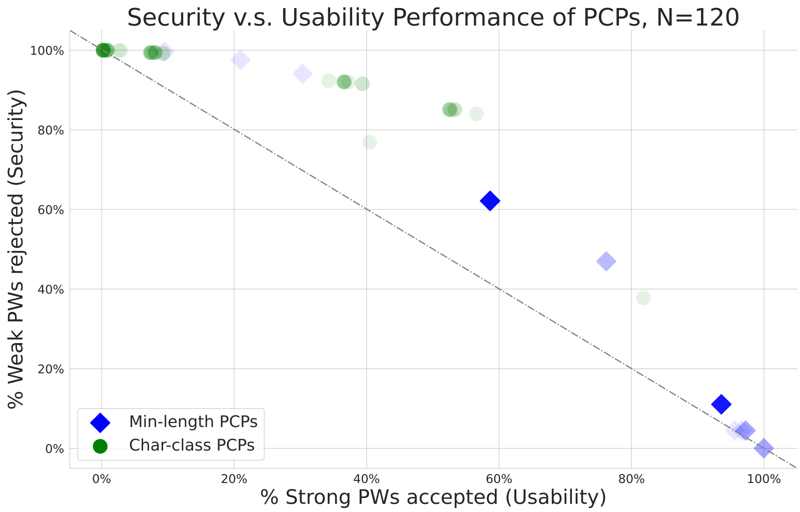


Figure 5.4.: Scatter plot of security vs usability of PCPs for 120 websites. Each data point was plotted with 10% opacity, so more opaque areas reflect higher concentrations of PCPs with close scores.

5.6.2. Results

Fig. 5.4 shows the scores of all websites we examined plotted along security and usability scores. Most of the 120 websites fall into one of three clusters: good security but poor usability (on the top left), good usability but poor security (bottom right), and average security and usability (in the middle of the graph). For comparison to a baseline, we also plotted a hypothetical PCP that rejects a random proportion α of passwords (and accepts $1 - \alpha$ of passwords), the diagonal line represents that PCP's security and usability scores for $0 \leq \alpha \leq 1$. Our findings are as follows:

1. **No PCP simultaneously had more than 60% security and usability.** They either rejected too many strong passwords or accepted too many weak ones. Note that a hypothetical random PCP that blocks 50% of passwords has 50% security and usability simultaneously.
2. **PCPs fall on different parts of the security-usability spectrum.** Our results suggest a classic security-usability tension among PCPs. 69 / 120 websites we

studied take opposite stances on the tradeoff; 33 have lenient policies (poor security cluster) and 36 have overly-stringent policies (poor usability cluster). Unsurprisingly, the PCPs within each of the 2 clusters are very similar to one another, with *1class6* being the majority PCP in the poor security cluster and *3class8* the majority in the poor usability cluster. We hypothesize that any PCP cannot be usable without allowing some weak passwords, and it cannot be secure without rejecting some strong password candidates.

1class8 policies make up most of the PCPs with middling acceptance rates, rejecting only 62% of weak passwords and accepting only 58% of strong passwords. While NIST recommends this exact composition policy, our results suggest that the PCP alone is insufficient in preventing users from choosing weak passwords; websites need to have additional safeguards—such as blocklists—to filter out the remaining 38%. This was not the case at 9 *1class8* websites, including SoundCloud, Eventbrite, and Trello; they allowed all of the *leaked* and *easiest-guessed* passwords we tried, like "`1234qwer`","`1234567890`", and "`babygirl123`" (cross-referencing our findings from § 5.3).

3. **Most websites with insecure PCPs do not prevent insecure password choices.** 22 of the 33 websites in the poor security cluster—including Amazon, Fox News, Etsy, and Dropbox (all with a *1class6* PCP)—do not block users from choosing passwords like "`abc123`" and "`qwerty`"—which we found with our password blocking analysis (§ 5.3)—and 2 more have insufficient blocking strategies (Slack and Yelp).

Our key finding is that PCPs are unsatisfactory in one or more ways. None of the 120 PCPs had more than 60% security and usability simultaneously. We hypothesize that there is no perfectly secure and usable PCP; all composition policies must make a tradeoff between user convenience (minimum-length PCPs) and strong passwords (character-class PCPs). Future studies should further investigate this hypothesis with different password corpora and methods. While websites with lenient PCPs can moderate the security gap with additional interventions like blocklists, we see this is not typically the case. A majority of these sites allow *leaked* and *easiest-guessed* passwords to be used.

5.7. Limitations

5.7.1. Limitations of analyzing the most popular websites

In these studies, we focused on the most popular websites. Since we did not additionally examine password policies of websites at the long tail (due to the work required to manually visit each website), we cannot be confident that our findings generalize to all websites. But note that previous research suggests that long-tail websites are likely to have even weaker security policies [124].

In § C.4, we detail the access failures encountered at 142 websites in the ranked list we used. While future research can make an effort to study some of their password policies (like at government websites), we don't believe their exclusion here affects our overall finding: most top websites are not following best practices in their password policies. Moreover, 83 / 142 excluded websites did not collect passwords, shared authentication with a website we already analyzed, or were unreachable (e.g., DNS, measurement links).

5.7.2. Limitations in the PCP security / usability analysis

In the PCP security / usability analysis, we rated the strength of all 10 million Xato passwords using PGS under no policy, which served as our ground truth. As PGS conservatively simulates an adversary cracking passwords, it also offers to configure guess number calculations under a particular PCP, since the adversary—who knows the website's PCP—can constrain their search space to guess passwords more efficiently (the default option is no policy). Uploaded passwords that were compliant with the selected policy (17 options at the time of writing) would then be guessed with modified approaches using each of the cracking tools [24]. Since we did not select a policy to use, our results may lead to slight overreportings of both the fraction of strong passwords accepted and the fraction of weak passwords rejected for some of the PCPs. Obtaining more accurate ground truth measures would be challenging: PGS limits submissions to 30,000 passwords in order to ensure fair use of their free service, and their cracking tools can take a few weeks to complete.[3]

We did, however, further investigate the ramifications of using the no-policy guess numbers in our security / usability analysis, and found our main findings still hold true. We found that the *false positive risk*—the probability that a password rated strong was actually weak—was less than 4.56% at the 10^6 guesses threshold we used, for all 120 PCPs.[4] We randomly sampled 30,000 compliant passwords—weighted by their frequency—for each PCP and obtained their "PCP-aware" guess numbers from PGS in order to make the pairwise comparisons.

5.7.3. Limitations in scale

Our study required a significant amount of manual work to learn all of the password policies. For example, in our blocklist analysis alone, we attempted 4,800 password changes to determine whether websites allowed *leaked* and *easiest-guessed* passwords (~200 hours of work). Since we manually visited each website to reverse-engineer

[3] The Passwords Research Team allowed us to submit all 10 million Xato passwords at once—for cracking under no policy—as a courtesy.

[4] Here we are concerned with the probability of a positive result being false [133]. This is different from the type 1 error rate (the false positive rate).

their password policy, we were only able to test 120 of the top English-language websites. We hence did not try to draw statistically valid conclusions about differences between industry sectors (e.g. news vs. social media websites) because of the small number of websites. We leave those topics (e.g., how the rates of compliance with best practices might vary by rank, geographic location, or sector) as future research directions.

We initially attempted two automated approaches which we ultimately abandoned due to concerns with completeness and data quality. We include our experiences in § C.2 to hopefully serve as useful notes for those who want to extend our work.

5.8. Other related work

Some previous empirical works have partially looked at password policies in practice. Bonneau and Preibusch (2010) extracted the PCPs of 150 websites across 3 different site categories: identity providers, content providers, and e-commerce sites [124]. They found that identity providers were significantly more likely to have minimum-length requirements (>1 character password), character-class requirements, and basic dictionary checks whenever a user changes their password. Overall, they found poor adoption of industry standards for password implementations, such as using TLS, CAPTCHA, and rate-limiting password guesses.

de Carnavalet et al. (2015) studied the password strength meters used at 11 popular websites [111]. They extracted or reverse-engineered the meter implementation at each site to local scripts and ran large-scale automated tests to get strength readings of known passwords, running a total of 53 million tests. They found most meters were only measuring password complexity, with only one implementation—`zxcvbn`—going beyond to penalize dictionary words. They also found that among the password strength meters in use on the web, most of them were inaccurate and inconsistent; passwords rated weak were often rated strong at other websites, and vice versa.

We built on the work done in both studies to deliver new additional insights. For password strength meters, we found websites with minimum-length PCPs that were using their strength meters as character-class nudges (§ 5.4). We also focused on investigating consistency between meter feedback at the client and password acceptance at the server by attempting to set the 20 *easiest-guessed* passwords we tested, and found more than half of websites were inconsistent. For PCPs, we resurveyed the landscape over a decade later, and found changes in the types of requirements used (§ 5.5). We also developed a new method to measure the security and usability of PCPs, and tentatively found none of them had decent security and usability simultaneously (§ 5.6).

Our findings further confirm little improvement in industry since the release of those previous studies. Character-class PCPs are still widely-used, and websites that use them are more likely to allow *leaked* and *easiest-guessed* passwords. We found a significant number of password strength meters still use complexity instead of guessability to represent strength, and only 23 websites have adopted strength meters of any kind.

5.9. Summary

Even with the gains in user authentication methods over the past two decades, passwords remain essential for online access, and replacing them in the near future seems improbable [72]. For these reasons, online services—especially the websites in which we found flaws—need to focus on password security and usability. Websites with insufficient blocklisting strategies, an outdated character-class PCP, or a misconfigured password strength meter should review the best practices summarized in Table 5.1 and make adjustments to their password policies. We further encourage them to review the research behind the guidelines in order to avoid misconfigured interventions that are inconsistent with one another (e.g., § 5.4).

We also suggest future research that directly engages with system administrators, in order to understand their mindset on password security. Researchers may then be able to uncover the reasons for the disconnect between industry and the academic community, and take steps towards reconciling the disparity. Some hypotheses include:

- Password policy is security theater: measures such as character-class PCPs, even if ineffective, may give users a false sense of security, and websites use them for this reason.
- Websites have shifted their attention to adopting other authentication technologies, such as multi-factor authentication (MFA), and believe that it is unnecessary to strengthen their password policies. (Note that there are severe weaknesses in SMS-based MFA, so this view might be overoptimistic [1, 2]).
- Websites need to pass security audits, and the firms who do these audits, such as Deloitte, recommend or mandate outdated practices.
- Websites face some other practical constraint that the academic community does not know about.

We have made our dataset available for other researchers at: `https://passwordpolicies.cs.princeton.edu/`.

6
Conclusion

In this dissertation, we aimed to discover weaknesses in user authentication practices. We found insecurity in practices that were not cutting-edge, but impacted the safety of millions of users. Moreover, the flaws present could easily be exploited by low-tech, UI-bound adversaries since they were not software flaws, but policy and process flaws. We made measurements to further find that many users were at risk, and have engaged in outreach to better protect the users.

These weaknesses collectively demonstrated an ongoing disconnect between user authentication research and practice. While practice has lagged behind research, research has also not paid attention to constraints in practice. As a result, these flawed policies—which have placed users in danger of account compromise, financial fraud, and online harassment—remained undiscovered for years before we studied them.

Our work more broadly aimed to bridge this research-practice gap. We demonstrated the value of studying security policies. We developed a methodology of reverse-engineering security policies to find weaknesses in user authentication practices, making measurements to quantifying the risks to users, and outreach to call for policy solutions. We used our methodology and found straightforward weaknesses in widely-deployed user authentication schemes: call center authentication for SIM swap requests, SMS-based authentication at online services, and password policies of top websites. We believe this approach can be adopted in future research to look for and mitigate other examples of the research-practice gap in user authentication, as well as in other aspects of information security.

We were principally driven by the desire to produce research that aligned with societal benefit; we studied user authentication practices because they are relevant to millions of people. While researching security policy goes against what is valued by the top academic conferences, we instead value the belief that our studies have significantly improved user safety. Looking ahead, we encourage others in the information security research community to reorient their research incentives: think about how much good their work can do for the world.

Bibliography

[1] Kevin Lee et al. “An Empirical Study of Wireless Carrier Authentication for SIM Swaps”. In: *Proceedings of the 16th Symposium On Usable Privacy and Security (SOUPS)*. USENIX Association, August 2020, pp. 61–79. URL: https://www.usenix.org/system/files/soups2020-lee.pdf (visited on 05/20/2022).

[2] Kevin Lee and Arvind Narayanan. “Security and Privacy Risks of Number Recycling at Mobile Carriers in the United States”. In: *Proceedings of the 2021 APWG Symposium on Electronic Crime Research (eCrime)*. Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 1–17. DOI: 10.1109/eCrime54498.2021.9738792.

[3] Kevin Lee, Sten Sjöberg, and Arvind Narayanan. “Password policies of most top websites fail to conform to best practices”. In: *Proceedings of the Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*. USENIX Association, 2022.

[4] Malte Möser et al. “An Empirical Analysis of Traceability in the Monero Blockchain”. In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018), pp. 143–163.

[5] Harry Kalodner et al. “Blocksci: Design and applications of a blockchain analysis platform”. In: *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*. USENIX Association, August 2020, pp. 2721–2738. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/kalodner (visited on 05/20/2022).

[6] Ben Kaiser et al. “Adapting Security Warnings to Counter Online Disinformation”. In: *Proceedings of the 30th USENIX Security Symposium (USENIX Security)*. USENIX Association, August 2021, pp. 1163–1180. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/kaiser (visited on 05/20/2022).

[7] Paul A. Grassi et al. *NIST Special Publication 800-63B Digital Authentication Guidelines. Authentication and Lifecycle Management*. June 22, 2017. DOI: 10.6028/NIST.SP.800-63b.

[8] Syed W. Shah and Salil S. Kanhere. “Recent Trends in User Authentication – A Survey”. In: *IEEE Access* 7 (2019), pp. 112505–112519. DOI: 10.1109/ACCESS.2019.2932400.

[9] David M’Raihi et al. *HOTP: An HMAC-Based One-Time Password Algorithm*. Tech. rep. 4226. RFC Editor, December 2005. 37 pp. DOI: 10.17487/RFC4226. URL: https://rfc-editor.org/rfc/rfc4226.txt.

[10] David M’Raihi et al. *TOTP: Time-Based One-Time Password Algorithm*. Tech. rep. 6238. RFC Editor, May 2011. 16 pp. DOI: 10.17487/RFC6238. URL: https://rfc-editor.org/rfc/rfc6238.txt.

[11] Bobby Allyn. *She Gets Calls And Texts Meant For Elon Musk. Some Are Pretty Weird*. NPR. May 21, 2020. URL: https://www.npr.org/2020/05/21/858155045/she-gets-calls-and-texts-meant-for-elon-musk-some-are-pretty-weird (visited on 01/04/2021).

[12] Robert McMillan. *He Thought His Phone Was Secure; Then He Lost $24 Million to Hackers*. The Wall Street Journal. November 8, 2019. URL: https://www.wsj.com/articles/he-thought-his-phone-was-secure-then-he-lost-24-million-to-hackers-11573221600 (visited on 12/01/2019).

[13] Adam Gabbatt. *Trump's Twitter hacked after Dutch researcher claims he guessed password – report*. The Guardian. October 22, 2020. URL: https://www.theguardian.com/us-news/2020/oct/22/trump-twitter-hacked-dutch-researcher-password (visited on 04/13/2022).

[14] Valerie Fanelle et al. "Blind and Human: Exploring More Usable Audio CAPTCHA Designs". In: *Proceedings of the 16th Symposium On Usable Privacy and Security (SOUPS)*. USENIX Association, August 2020, pp. 111–125. URL: https://www.usenix.org/conference/soups2020/presentation/fanelle (visited on 05/16/2022).

[15] Rahul Chatterjee et al. "The Spyware Used in Intimate Partner Violence". In: *Proceedings of the 39th IEEE Symposium on Security & Privacy (S&P)*. Institute of Electrical and Electronics Engineers (IEEE), May 2018, pp. 441–458. DOI: 10.1109/SP.2018.00061.

[16] Neustar. *2021 State of Call Center Authentication*. URL: https://www.cdn.neustar/resources/whitepapers/risk/neustar-state-of-call-center-authentication-2021.pdf (visited on 05/13/2022).

[17] Dave Childers. *State of the Auth 2021: Experiences and Perceptions of Multi-Factor Authentication*. Tech. rep. Duo Labs, 2021. URL: https://duo.com/assets/ebooks/state-of-the-auth-2021.pdf.

[18] Alex Stamos. "Tackling the Trust and Safety Crisis". In: USENIX Association, 2019. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/stamos.

[19] James Mickens. "This World of Ours". In: *;login: logout* (2014). URL: https://www.usenix.org/system/files/1401_08-12_mickens.pdf.

[20] Dror G. Feitelson. *Experimental Computer Science: The Need for a Cultural Change*. Tech. rep. The Hebrew University of Jerusalem, 2006. URL: https://www.cs.huji.ac.il/w~feit/papers/exp05.pdf.

[21] Diana Freed et al. ""A Stalker's Paradise": How Intimate Partner Abusers Exploit Technology". In: *Proceedings of the 2018 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), April 2018, pp. 1–13. DOI: 10.1145/3173574.3174241.

[22] Federal Communications Commission. *In the Matter of Protecting Consumers from SIM Swap and Port- Out Fraud. Notice of Proposed Rulemaking*. URL: https://docs.fcc.gov/public/attachments/FCC-21-102A1.pdf (visited on 05/04/2022).

[23] Arvind Narayanan and Bendert Zevenbergen. *No Encore for Encore? Ethical questions for web-based censorship measurement*. Tech. rep. Council for Big Data, Ethics, and Society, 2015. URL: https://bdes.datasociety.net/wp-content/uploads/2016/10/Encore-Case-Study.pdf.

[24] Blase Ur et al. "Measuring Real-World Accuracies and Biases in Modeling Password Guessability". In: *Proceedings of the 24th USENIX Security Symposium (USENIX Security)*. USENIX Association, August 2015, pp. 463–481. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/ur.

[25] Kevin Lee et al. *Vulnerability reporting is dysfunctional*. Freedom to Tinker. March 25, 2020. URL: https://freedom-to-tinker.com/2020/03/25/vulnerability-reporting-is-dysfunctional/ (visited on 04/27/2022).

[26] Daehyun Strobel. "IMSI catcher". MA thesis. Ruhr-Universität Bochum, July 13, 2007. URL: https://www.emsec.ruhr-uni-bochum.de/media/

crypto/attachments/files/2011/04/imsi_catcher.pdf (visited on 06/08/2020).

[27] Chris Paget. "Practical cellphone spying". 31st Chaos Communication Congress (31C3). August 2010. URL: http://index-of.es/Miscellanous/LIVRES/cellphonespying.pdf (visited on 06/08/2020).

[28] Altaf Shaik et al. *Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems*. August 7, 2017.

[29] Byeongdo Hong, Sangwook Bae, and Yongdae Kim. "GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier". In: *Proceedings of the 25th Network and Distributed System Security Symposium (NDSS)*. Internet Society (ISOC), February 2018. DOI: 10.14722/ndss.2018.23349.

[30] Tyler Moore et al. "Signaling system 7 (SS7) network security". In: *The 2002 45th Midwest Symposium on Circuits and Systems (MWSCAS) Conference Proceedings*. Institute of Electrical and Electronics Engineers (IEEE), August 2002, pp. 496–499. DOI: 10.1109/MWSCAS.2002.1187082.

[31] Positive Technologies. *SS7 Security Report*. URL: https://positive-tech.com/storage/articles/ss7-security-report-2014-eng.pdf (visited on 06/07/2020).

[32] Karsten Nohl. "Mobile self-defense". 31st Chaos Communication Congress (31C3). December 27, 2014. URL: https://fahrplan.events.ccc.de/congress/2014/Fahrplan/system/attachments/2493/original/Mobile_Self_Defense-Karsten_Nohl-31C3-v1.pdf (visited on 06/08/2020).

[33] Lily Hay Newman. *Fixing the Cell Network Flaw That Lets Hackers Drain Bank Accounts*. WIRED. May 9, 2017. URL: https://www.wired.com/2017/

05/fix-ss7-two-factor-authentication-bank-accounts/ (visited on 12/01/2019).

[34] Silke Holtmanns and Ian Oliver. “SMS and one-time-password interception in LTE networks”. In: *2017 IEEE International Conference on Communications(ICC 2017)*. Institute of Electrical and Electronics Engineers (IEEE), May 2017, pp. 5711–5716. DOI: 10.1109/ICC.2017.7997246.

[35] Joseph Bonneau et al. “Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google”. In: *Proceedings of the 24th World Wide Web Conference (WWW)*. International World Wide Web Conference Committee (IW3C2), May 2015, pp. 141–150. DOI: 10.1145/2736277.2741691.

[36] Jessica Colnago et al. ““It’s not actually that horrible”: Exploring Adoption of Two-Factor Authentication at a University”. In: *Proceedings of the 2018 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), April 2018, pp. 1–11. DOI: 10.1145/3173574.3174030.

[37] Catherine S Weir et al. “User perceptions of security, convenience and usability for ebanking authentication tokens”. In: *Computers and Security* 28 (1–2 2009), pp. 47–62. DOI: 10.1016/j.cose.2008.09.008.

[38] Elissa M. Redmiles, Michelle L. Mazurek, and John P. Dickerson. “Dancing Pigs or Externalities?: Measuring the Rationality of Security Decisions”. In: *EC ’18: Proceedings of the 2018 ACM Conference on Economics and Computation*. Association for Computing Machinery (ACM), June 2018, pp. 215–232. DOI: 10.1145/3219166.3219185.

[39] Ronald Campbell. *Your number is up!* The Orange County Register. October 15, 1998. URL: `https://web.archive.org/web/19990222023921/http://www.ocregister.com/business/codex105w.shtml` (visited on 04/04/2021).

[40] North American Numbering Plan Administrator. *October 2020 North American Numbering Plan (NANP) Exhaust Analysis*. October 2020. URL: `https://www.nationalnanpa.com/reports/October_2020_NANP_Exhaust_AnalysisFinal.pdf` (visited on 04/04/2021).

[41] Federal Communications Commission. *In the Matter of Advanced Methods to Target and Eliminate Unlawful Robocalls. Second Notice of Inquiry*. URL: `https://docs.fcc.gov/public/attachments/FCC-17-90A1.pdf` (visited on 01/18/2021).

[42] Federal Communications Commission. *In the Matter of Advanced Methods to Target and Eliminate Unlawful Robocalls. Second Report and Order*. URL: `https://docs.fcc.gov/public/attachments/FCC-18-177A1.pdf` (visited on 01/18/2021).

[43] Federal Communications Commission. *Consumer and Governmental Affairs Bureau Releases Report to Congress on the Status of the Reassigned Numbers Database*. URL: `https://docs.fcc.gov/public/attachments/DOC-368620A1.pdf` (visited on 01/18/2021).

[44] Joseph Bonneau. “The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords”. In: *Proceedings of the 33rd IEEE Symposium on Security & Privacy (S&P)*. Institute of Electrical and Electronics Engineers (IEEE), May 2012, pp. 538–552. DOI: `10.1109/SP.2012.49`.

[45] Matt Weir et al. “Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords”. In: *Proceedings of the 17th ACM SIGSAC*

Conference on Computer and Communications Security (CCS). Association for Computing Machinery (ACM), October 2010, pp. 162–175. DOI: 10.1145/1866307.1866327.

[46] William Burr et al. *NIST Special Publication 800-63-2 Electronic Authentication Guideline*. August 2013. DOI: 10.6028/NIST.SP.800-63-2.

[47] Matteo Dell'Amico, Pietro Michiardi, and Yves Roudier. "Password Strength: An Empirical Analysis". In: *2010 Proceedings IEEE INFOCOM*. Institute of Electrical and Electronics Engineers (IEEE), March 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5461951.

[48] Brian Barrett. *How to Protect Yourself Against a SIM Swap Attack*. WIRED. August 19, 2018. URL: https://www.wired.com/story/sim-swap-attack-defend-phone/ (visited on 12/01/2019).

[49] Brian Krebs. *Busting SIM Swappers and SIM Swap Myths*. Krebs on Security. November 7, 2018. URL: https://krebsonsecurity.com/2018/11/busting-sim-swappers-and-sim-swap-myths/ (visited on 12/01/2019).

[50] Lorenzo Franceschi-Bicchierai. *How Criminals Recruit Telecom Employees to Help Them Hijack SIM Cards*. Motherboard. August 3, 2018. URL: https://www.vice.com/en_us/article/3ky5a5/criminals-recruit-telecom-employees-sim-swapping-port-out-scam (visited on 12/01/2019).

[51] Paul A. Grassi et al. *DRAFT NIST Special Publication 800-63B Digital Authentication Guidelines. Authentication and Lifecycle Management*. June 24, 2016. URL: https://web.archive.org/web/20160624033024/https://pages.nist.gov/800-63-3/sp800-63b.html (visited on 02/15/2019).

[52] Better Business Bureau of Central Oklahoma. *BBB Warns About Cell Phone Porting Scams*. February 6, 2018. URL: https://www.bbb.org/article/news-

releases/17019-bbb-warns-about-cell-phone-porting-scams (visited on 12/01/2019).

[53] PCI Security Standards Council, LLC. *Payment Card Industry (PCI) Data Security Standard. Requirements and Security Assessment Procedures Version 3.2.1*. URL: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf?agreement=true&time=1591585370712 (visited on 06/07/2020).

[54] John Podd, Julie Bunnell, and Ron Henderson. "Cost-Effective Computer Security: Cognitive and Associative Passwords". In: *Proceedings of the Sixth Australian Conference on Computer-Human Interaction (OZCHI)*. Institute of Electrical and Electronics Engineers (IEEE), November 1996, pp. 304–305. DOI: 10.1109/OZCHI.1996.560026.

[55] Stuart Schechter, A. J. Bernheim Brush, and Serge Egelman. "It's No Secret. Measuring the Security and Reliability of Authentication via "Secret" Questions". In: *Proceedings of the 30th IEEE Symposium on Security & Privacy (S&P)*. Institute of Electrical and Electronics Engineers (IEEE), May 2009, pp. 375–390. DOI: 10.1109/SP.2009.11.

[56] Lorrie Cranor. *Your mobile phone account could be hijacked by an identity thief*. Tech@FTC. June 7, 2016. URL: https://www.ftc.gov/news-events/blogs/techftc/2016/06/your-mobile-phone-account-could-be-hijacked-identity-thief (visited on 06/07/2020).

[57] Federal Trade Commission. *Consumer Sentinel Network Data Book for January – December 2015*. URL: https://www.ftc.gov/system/files/documents/reports/consumer-sentinel-network-data-book-january-december-2015/160229csn-2015databook.pdf (visited on 06/08/2020).

[58] Action Fraud. *Alert – how you can be scammed by a method called SIM Splitting*. May 9, 2014. URL: https://www.actionfraud.police.uk/alert/alert-how-you-can-be-scammed-by-a-method-called-sim-splitting (visited on 12/01/2019).

[59] Mateen Hafeez. *SIM fraud: Police zero in on public phone booth owners*. The Times of India. August 9, 2008. URL: https://timesofindia.indiatimes.com/city/mumbai/SIM-fraud-Police-zero-in-on-public-phone-booth-owners/articleshow/3344515.cms (visited on 12/01/2019).

[60] Clayton Barnes. *Beware SIM card swop scam*. The Saturday Star. January 7, 2008. URL: https://www.security.co.za/news/5907 (visited on 12/01/2019).

[61] Jason Aten. *SIM Swapping Is the Biggest Security Threat You Face, and Almost No One Is Trying to Fix It. Here's Why It Matters*. Inc. September 17, 2019. URL: https://www.inc.com/jason-aten/sim-swapping-is-one-of-biggest-cyber-security-threats-you-face-almost-no-one-is-trying-to-fix-it-heres-why-it-matter.html (visited on 12/01/2019).

[62] Lorenzo Franceschi-Bicchierai. *T-Mobile Has a Secret Setting to Protect Your Account From Hackers That It Refuses to Talk About*. Motherboard. September 13, 2019. URL: https://www.vice.com/en_us/article/ywa3dv/t-mobile-has-a-secret-setting-to-protect-your-account-from-hackers-that-it-refuses-to-talk-about (visited on 01/06/2020).

[63] Verizon Wireless. *Transfer (port out) your number to another carrier FAQs. Number Lock*. URL: https://www.verizonwireless.com/support/port-out-faqs/ (visited on 04/06/2020).

[64] Frost & Sullivan TEAM Research. *Consumer Communication Services Tracker, Q3 2019*. Frost & Sullivan. July 5, 2019. URL: https://store.frost.com/

consumer-communication-services-tracker-q3-2019.html (visited on 06/07/2020).

[65] T-Mobile US, Inc. *Q3 2019. Financial Results, Supplementary Data, Non-GAAP Reconciliations, Reconciliation of Operating Measures*. URL: https://s22.q4cdn.com/194431217/files/doc_financials/2019/q3/TMUS-09_30_2019-Financial-Results,-Supplemental-Data,-Non-GAAP-Reconciliations,-and-reconciliation-of-operating-measures-FINAL.pdf (visited on 06/07/2020).

[66] Verizon Communications. *2018 Annual Report*. URL: https://www.verizon.com/about/sites/default/files/2018-Verizon-Annual-Report.pdf (visited on 06/07/2020).

[67] Nitrokey. *DongleAuth.info*. URL: https://www.dongleauth.info/ (visited on 06/07/2020).

[68] Elie Bursztein. *The bleak picture of two-factor authentication adoption in the wild*. Elie. December 21, 2018. URL: https://elie.net/blog/security/the-bleak-picture-of-two-factor-authentication-adoption-in-the-wild/ (visited on 06/07/2020).

[69] HackerOne. *Improving Public Bug Bounty Programs with Signal Requirements*. HackerOne Blog. March 15, 2016. URL: https://www.hackerone.com/blog/signal-requirements (visited on 06/07/2020).

[70] Aron Laszka, Mingyi Zhao, and Jens Grossklags. "Banishing Misaligned Incentives for Validating Reports in Bug-Bounty Platforms". In: *Computer Security – European Symposium on Research in Computer Security (ESORICS) 2016*. Vol. 9879. Lecture Notes in Computer Science (LNCS). Springer Inter-

national Publishing, September 2016, pp. 161–178. doi: 10.1007/978-3-319-45741-3_9.

[71] Gartner, Inc. *Duo Security Competitors and Alternatives in User Authentication Reviews*. url: https://www.gartner.com/reviews/market/user-authentication/vendor/duo-security/alternatives (visited on 02/25/2020).

[72] Joseph Bonneau et al. "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes". In: *Proceedings of the 33rd IEEE Symposium on Security & Privacy (S&P)*. Institute of Electrical and Electronics Engineers (IEEE), May 2012, pp. 553–567. doi: 10.1109/SP.2012.44.

[73] Joseph Bonneau, Sören Preibusch, and Ross Anderson. "A Birthday Present Every Eleven Wallets? The Security of Customer-Chosen Banking PINs". In: *Proceedings of the 16th International Conference on Financial Cryptography and Data Security (FC)*. Vol. 7397. Lecture Notes in Computer Science (LNCS). Springer International Publishing, March 2012, pp. 25–40. doi: 10.1007/978-3-642-32946-3_3.

[74] Linda Serges. *Q3 2018 Mobile Trade In Data: The iPhone Effect*. Hyla Blog. October 25, 2018. url: https://blog.hylamobile.com/q3-2018-mobile-trade-in-data-the-iphone-effect (visited on 12/01/2019).

[75] Rebecca Balebako et al. "The Privacy and Security Behaviors of Smartphone App Developers". In: *Proceedings of the Workshop on Usable Security (USEC)*. February 2014. doi: 10.14722/usec.2014.23006.

[76] Peter Leo Gorski et al. "Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic API Misuse". In: *Proceedings of the 14th Symposium On Usable Privacy and Security (SOUPS)*.

USENIX Association, August 2018, pp. 265–281. URL: https://www.usenix.org/system/files/conference/soups2018/soups2018-gorski.pdf (visited on 06/08/2020).

[77] Ken Reese et al. "A Usability Study of Five Two-Factor Authentication Methods". In: *Proceedings of the 15th Symposium On Usable Privacy and Security (SOUPS)*. USENIX Association, August 2019, pp. 357–370. URL: https://www.usenix.org/system/files/soups2019-reese.pdf (visited on 06/08/2020).

[78] Pew Research Center. *Mobile Fact Sheet*. Pew Research Center: Internet, Science & Tech. June 12, 2019. URL: https://www.pewresearch.org/internet/fact-sheet/mobile/ (visited on 06/07/2020).

[79] Brian Krebs. *Why Phone Numbers Stink As Identity Proof*. Krebs on Security. March 17, 2019. URL: https://krebsonsecurity.com/2019/03/why-phone-numbers-stink-as-identity-proof/ (visited on 01/04/2021).

[80] Nancy Lloyd. *Why giving up your phone number can mean giving up your privacy*. Los Angeles Times. November 26, 2016. URL: https://www.latimes.com/business/la-fi-tn-phone-number-security-20161125-story.html (visited on 01/08/2021).

[81] Allison McDonald et al. "The Annoying, the Disturbing, and the Weird: Challenges with Phone Numbers as Identifiers and Phone Number Recycling". In: *Proceedings of the 2021 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), May 2021. DOI: 10.1145/3411764.3445085.

[82] 2factorauth. *2FA Directory*. URL: https://2fa.directory/ (visited on 01/15/2022).

[83] Federal Communications Commission. *Numbering Resource Utilization in the United States*. URL: `https://docs.fcc.gov/public/attachments/DOC-367592A1.pdf` (visited on 01/18/2021).

[84] Philip Bump. *People are paying tens of thousands of dollars for custom phone numbers. These are the most expensive*. The Washington Post. April 23, 2015. URL: `https://www.washingtonpost.com/news/wonk/wp/2015/04/23/how-to-make-100000-by-selling-a-phone-number-on-the-internet/` (visited on 01/09/2021).

[85] Google. *Google Voice Acceptable Use Policy*. URL: `https://www.google.com/googlevoice/program-policies.html` (visited on 04/15/2021).

[86] Twilio. *Best Practices for Phone Number Use*. URL: `https://www.twilio.com/docs/phone-numbers/best-practices` (visited on 04/15/2021).

[87] Federal Communications Commission. *Numbering Resources*. URL: `https://www.fcc.gov/general/numbering-resources` (visited on 01/14/2021).

[88] Federal Communications Commission. *Consumer and Governmental Affairs Bureau Announces Compliance Date for Reassigned Numbers Database Rules*. URL: `https://docs.fcc.gov/public/attachments/DA-20-706A1.pdf` (visited on 01/18/2021).

[89] Federal Communications Commission. *Reassigned Numbers Database*. URL: `https://www.fcc.gov/reassigned-numbers-database` (visited on 01/23/2022).

[90] Federal Communications Commission. *Reassigned Numbers Database (RND) Technical Requirements Document*. URL: `https://docs.fcc.gov/public/attachments/DOC-361954A1.pdf` (visited on 01/18/2021).

[91] Linus Särud. *The danger of recycled phone numbers*. Detectify Labs. May 24, 2018. URL: https://labs.detectify.com/2018/05/24/recycled-phone-numbers/ (visited on 01/04/2021).

[92] Verizon Media. *Verizon Media Terms of Service*. February 2021. URL: https://www.verizonmedia.com/policies/us/en/verizonmedia/terms/otos/index.html (visited on 04/15/2021).

[93] Microsoft. *Microsoft Services Agreement*. August 1, 2020. URL: https://www.microsoft.com/en-us/servicesagreement/default.aspx (visited on 04/15/2021).

[94] Google. *Create a replacement Google Account*. URL: https://support.google.com/accounts/answer/7564124 (visited on 04/15/2021).

[95] Doug Gross. *Yahoo 'recycling' old e-mail, raising security concerns*. CNN. June 20, 2013. URL: https://www.cnn.com/2013/06/20/tech/web/yahoo-recycled-email/index.html (visited on 01/19/2021).

[96] Jack Schofield. *Hotmail: are my lost accounts a security risk?* The Guardian. July 18, 2013. URL: https://www.theguardian.com/technology/askjack/2013/jul/18/hotmail-lost-accounts-security-risk (visited on 01/19/2021).

[97] Federal Trade Commission. *How to Recognize and Report Spam Text Messages*. August 31, 2020. URL: https://www.consumer.ftc.gov/articles/how-recognize-and-report-spam-text-messages (visited on 01/20/2021).

[98] Lily Hay Newman. *ShinyHunters Is a Hacking Group on a Data Breach Spree*. WIRED. May 21, 2020. URL: https://www.wired.com/story/shinyhunters-hacking-group-data-breach-spree/ (visited on 02/08/2021).

[99] Brian Krebs. *The Market for Stolen Account Credentials*. Krebs on Security. December 18, 2017. URL: `https://krebsonsecurity.com/2017/12/the-market-for-stolen-account-credentials/` (visited on 01/11/2021).

[100] Riva Richmond. *Stolen Facebook Accounts for Sale*. The New York Times. May 2, 2010. URL: `https://www.nytimes.com/2010/05/03/technology/internet/03facebook.html` (visited on 01/11/2021).

[101] Noah Kelley. *DIY Cybersecurity for Domestic Violence. My partner is harassing me through my cell phone.* HACK*BLOSSOM. URL: `https://hackblossom.org/domestic-violence/threats/cell-phones.html` (visited on 03/21/2021).

[102] Brian Willingham. *Intelius vs. Spokeo vs. BeenVerified — A Private Investigator's Review*. May 8, 2019. URL: `https://diligentiagroup.com/background-investigations/intelius-vs-spokeo-vs-been-verified-private-investigator-review/` (visited on 03/22/2021).

[103] Rachael Clemmons. *Intelius Review: Is the Background Check Service Worth It?* November 7, 2019. URL: `https://www.asecurelife.com/intelius-review/` (visited on 02/25/2021).

[104] Sathvik Prasad et al. "Who's Calling? Characterizing Robocalls through Audio and Metadata Analysis". In: *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*. USENIX Association, August 2020, pp. 397–414. URL: `https://www.usenix.org/system/files/sec20-prasad.pdf` (visited on 04/15/2021).

[105] Michelle Caruthers. *World Password Day: How to Improve Your Passwords*. May 18, 2018. URL: `https://blog.dashlane.com/world-password-day/` (visited on 01/18/2021).

[106] Adam Fendelman. *How to Park Your Cell Phone Number*. Lifewire. November 14, 2019. URL: https://www.lifewire.com/parking-how-to-hold-cell-number-577582 (visited on 04/04/2021).

[107] Yael Grauer. *How to Delete Your Information From People-Search Sites*. August 20, 2020. URL: https://www.consumerreports.org/personal-information/how-to-delete-your-information-from-people-search-sites/ (visited on 03/23/2021).

[108] Patrick Gage Kelley et al. “Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms”. In: *Proceedings of the 33rd IEEE Symposium on Security & Privacy (S&P)*. Institute of Electrical and Electronics Engineers (IEEE), May 2012, pp. 523–537. DOI: 10.1109/SP.2012.38.

[109] Joshua Tan et al. “Practical Recommendations for Stronger, More Usable Passwords Combining Minimum-Strength, Minimum-Length, and Blocklist Requirements”. In: *Proceedings of the 27th ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery (ACM), October 2020, pp. 1407–1426. DOI: 10.1145/3372297.3417882.

[110] Dan U. *Passwords, passwords everywhere*. National Cyber Security Centre. April 21, 2019. URL: https://www.ncsc.gov.uk/blog-post/passwords-passwords-everywhere (visited on 12/21/2021).

[111] Xavier De Carné De Carnavalet and Mohammad Mannan. “From Very Weak to Very Strong: Analyzing Password-Strength Meters”. In: *Proceedings of the 21st Network and Distributed System Security Symposium (NDSS)*. Internet Society (ISOC), February 2014. DOI: 10.14722/ndss.2014.23268.

[112] Blase Ur et al. “Design and Evaluation of a Data-Driven Password Meter”. In: *Proceedings of the 2017 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), April 2017, pp. 3775–3786. DOI: 10.1145/3025453.3026050.

[113] Richard Shay et al. “A Spoonful of Sugar? The Impact of Guidance and Feedback on Password-Creation Behavior”. In: *Proceedings of the 2015 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), April 2015, pp. 2903–2912. DOI: 10.1145/2702123.2702586.

[114] William Melicher et al. “Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks”. In: *Proceedings of the 25th USENIX Security Symposium (USENIX Security)*. USENIX Association, August 2016, pp. 175–191. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/melicher.

[115] Dinei Florêncio, Cormac Herley, and Paul C. van Oorschot. “An Administrator’s Guide to Internet Password Research”. In: *Proceedings of the28th Large Installation System Administration Conference (LISA14)*. USENIX Association, November 2014, pp. 44–61. URL: https://www.usenix.org/conference/lisa14/conference-program/presentation/florencio.

[116] Saranga Komanduri et al. “Of Passwords and People: Measuring the Effect of Password-Composition Policies”. In: *Proceedings of the 2011 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), April 2011, pp. 2595–2604. DOI: 10.1145/1978942.1979321. URL: https://doi.org/10.1145/1978942.1979321.

[117] Verizon DBIR Team. *2020 Data Breach Investigations Report*. May 19, 2020. URL: https://www.verizon.com/business/resources/reports/dbir/2020/ (visited on 03/22/2022).

[118] ESET. *ESET Threat Report T3 2021*. February 9, 2022. URL: https://www.welivesecurity.com/wp-content/uploads/2022/02/eset_threat_report_t32021.pdf (visited on 03/22/2022).

[119] Steven Furnell. *Stop blaming people for choosing bad passwords – it's time websites did more to help*. The Conversation. January 3, 2022. URL: https://theconversation.com/stop-blaming-people-for-choosing-bad-passwords-its-time-websites-did-more-to-help-172257 (visited on 01/26/2022).

[120] Hana Habib et al. "Password Creation in the Presence of Blacklists". In: *Proceedings of the 2017 Workshop on Usable Security (USEC)*. February 2017. DOI: 10.14722/usec.2017.23043.

[121] Richard Shay et al. "Can Long Passwords Be Secure and Usable?" In: *Proceedings of the 2014 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), April 2014, pp. 2927–2936. DOI: 10.1145/2556288.2557377.

[122] Richard Shay et al. "Encountering Stronger Password Requirements: User Attitudes and Behaviors". In: *Proceedings of the 6th Symposium On Usable Privacy and Security (SOUPS)*. Association for Computing Machinery (ACM), July 2010, pp. 1–20. DOI: 10.1145/1837110.1837113.

[123] Blase Ur et al. ""I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab". In: *Proceedings of the 11th Symposium On Usable Privacy and Security (SOUPS)*. USENIX Association, July 2015, pp. 123–140.

URL: https://www.usenix.org/conference/soups2015/proceedings/presentation/ur.

[124] Joseph Bonneau and Sören Preibusch. “The password thicket: technical and market failures in human authentication on the web”. In: *The Ninth Workshop on the Economics of Information Security*. June 2010. URL: https://econinfosec.org/archive/weis2010/papers/session3/weis2010_bonneau.pdf.

[125] Daniel Lowe Wheeler. “zxcvbn: Low-Budget Password Strength Estimation”. In: *Proceedings of the 25th USENIX Security Symposium (USENIX Security)*. USENIX Association, August 2016, pp. 157–173. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/wheeler.

[126] Victor Le Pochat et al. “Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation”. In: *Proceedings of the 26th Network and Distributed System Security Symposium (NDSS)*. Internet Society (ISOC), February 2019. DOI: 10.14722/ndss.2019.23386.

[127] Mark Burnett. *Today I Am Releasing Ten Million Passwords*. February 9, 2015. URL: https://xato.net/today-i-am-releasing-ten-million-passwords-b6278bbe7495 (visited on 01/06/2022).

[128] Troy Hunt. *Introducing 306 Million Freely Downloadable Pwned Passwords*. August 3, 2017. URL: https://www.troyhunt.com/introducing-306-million-freely-downloadable-pwned-passwords/ (visited on 12/22/2021).

[129] Troy Hunt. *The 773 Million Record "Collection #1" Data Breach*. January 17, 2019. URL: https://www.troyhunt.com/the-773-million-record-collection-1-data-reach/ (visited on 12/22/2021).

[130] Blase Ur et al. "Do Users' Perceptions of Password Security Match Reality?" In: *Proceedings of the 2016 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Association for Computing Machinery (ACM), May 2016, pp. 3748–3760. DOI: `10.1145/2858036.2858546`.

[131] Apple. *Password Rules Validation Tool*. URL: `https://developer.apple.com/password-rules/` (visited on 02/14/2022).

[132] Mark Burnett. *Ten Million Passwords FAQ*. February 10, 2015. URL: `https://xato.net/ten-million-passwords-faq-3b2752ed3b4c` (visited on 01/06/2022).

[133] David Colquhoun. "The reproducibility of research and the misinterpretation of p-values". In: *Royal Society Open Science* 4 (12 December 2017). DOI: `10.1098/rsos.171085`.

A

Appendix to Chapter 3

A.1. Authentication for postpaid accounts

Table A.1.: Authentication methods we observed at each postpaid carrier. A checkmark means that a type of information was a component of at least one pathway for SIM swap customer authentication; it does not mean that a type of information was necessary or by itself sufficient for SIM swap customer authentication.

	Account Information	Device Information		Usage Information	Knowledge	Possession
	Account Number	IMEI	ICCID	Recent Numbers	PIN or Password	SMS OTP*
AT&T	●	●	●	●	●	●
T-Mobile					●	●
Verizon					●	

*We represent SMS OTP as a secure authentication factor because 1) we assume that a carrier sends the SMS OTP exclusively over its own network as a service message, such that the passcode is not vulnerable to routing attacks, and 2) we assume that if an attacker already has the ability to hijack a victim's SMS, a SIM swap does not provide the attacker with additional capabilities.

🟩	generally accepted in the computer security research field
🟥	had not been previously tested but we demonstrate is insecure (for reasons explained in § 3.5)
🟨	known to have security shortcomings (also for reasons described in § 3.5)

After completing our data collection on prepaid accounts, engaging with industry stakeholders, and reviewing public disclosures about wireless carrier account security, it appeared likely that authentication practices for postpaid accounts differed from authentication practices for prepaid accounts. We therefore followed our study of prepaid accounts with a study of postpaid accounts at 3 carriers: AT&T, T-Mobile, and Verizon.

We used a similar method for studying the postpaid carriers. Rather than using generated identities, members of the research team signed up with their own credentials. This was to address the additional identify verification process present at postpaid signups. We used the same threat model and script; after one week of usage we called in to request a SIM swap. To the best of our ability, we enabled all available safeguards against SIM swaps at each carrier by configuring our online profiles and calling in soon after to request protections against SIM swaps.[1]

It is important to note that postpaid accounts require real-world identities. Ultimately, we were only able to sign up for one account per carrier using the identities of research personnel. Therefore, the results of this study of postpaid carriers should be interpreted anecdotally. Spotting an authentication factor in this very limited run is some evidence that it is a component of the carrier's customer authentication flow, but not spotting an authentication factor provides little information. In other words, we believe these results are best interpreted as somewhat unlikely to include false positives for authentication factors, but we cannot offer much confidence about false negatives.

The calls were made in December 2019. Our IRB application was submitted in September 2019 and approved in November 2019. Results of our findings are shown in Table A.1.

[1] We also enabled the `NOPORT` option for T-Mobile, though our understanding is that the option only applies to port outs and not SIM swaps at present. Our understanding is also that T-Mobile does have additional protections against SIM swaps that can be associated with an account, but only after the account has been the victim of fraud.

A.2. Ethical considerations

Working with our institution's IRB, we took steps to minimize the risk of harm to both research personnel and customer service representatives, primarily by protecting their privacy.

A.2.1. Minimizing the risk of harm to RAs

We took steps to protect the privacy of the research assistants we hired. During account setup, we were required to provide the name of the account owner. Since prepaid accounts do not require a real-world identity, our protocol allowed RAs to use a fictitious name on the account if they elected for it. We assigned names using an online name generator.

The accounts were at all times controlled by the research team, and only the RA who had been designated as the account owner would be allowed to view information on that account. That is, RAs were not allowed access to accounts assigned to other RAs. The accounts were funded through the duration of the study and closed at the end of the experiment.

A.2.2. Minimizing the risk of harm to CSRs

We took two preventive measures to minimize the risk of harm to the customer service representatives who handled our calls:

- **Calls were not recorded.** The study design was approved with the parameter that the study procedures not be recorded due to differing laws regarding recordings across the states. Instead, we took detailed notes about the carrier's policies and practices during the call. Our notes do not include references to time of conversation (timestamps), gender, or any other identifying information related to the CSRs.

- **Account information will remain unpublished.** We have not revealed the phone numbers used in our study in order to minimize risk to CSRs. Otherwise, carriers would be able to track the service history on the accounts and potentially subject pertinent CSRs to disciplinary action (which would also be orthogonal to our study, since our research was designed to obtain information about corporate policies rather than about individuals).

We did not obtain the CSRs' informed consent before interacting with them, because our mitigations listed above ensure that the risks to them are minimal; they are simply carrying out their ordinary responsibilities. Furthermore, our study could not have been conducted with informed consent; firms might decline to participate or misrepresent their policies and practices. We obtained a waiver of consent from the IRB before carrying out our study. The Common Rule specifies a set of criteria for waiver, which we addressed in our IRB application.[2] While we did not debrief CSRs immediately after each SIM swap request, we provided an initial notification of our findings to the carriers we studied and to CTIA in July 2019 (even though our IRB did not impose an ex-post disclosure requirement).

A.3. Website responses to vulnerability reports

In early January 2020, we attempted to notify each of the 17 websites described in § 3.7.3 of the presence of doubly insecure configurations. We aimed to make as clear as possible the fact that our report was not merely a recapitulation of the already widely known possibility of SIM swaps and that our report was specific to

[2] 45 C.F.R. § 46.116(f).

Table A.2.: Responses from our vulnerability disclosure, detailed in § 3.7.3. Contacted platforms are in italicized font. Only in four of the 17 cases did the process work as expected, resulting in fixes.

*Email address not publicly available, we were provided the address only after sending a Twitter direct message (DM) asking for a reporting address.
**Zoho claims that its current policy—which disallows the same number to be used for recovery and MFA—is secure and does not require any changes.

Website	Available platforms	Response(s)	Default configuration	Days to fix
Adobe	*Security email*, HackerOne	Reported as fixed	Secure	10
Amazon	*Security email*	Closed as won't fix	Doubly insecure	—
Aol (Verizon Media)	*Security email*	No response	Doubly insecure	—
Blizzard	*Security email*	Template acknowledgment; later fixed without reporting	Doubly insecure	—
eBay	*Internal bug bounty*	Reported as fixed	Secure	28
Finnair	*Customer support portal*	No response	Doubly insecure	—
Gaijin Entertainment	*Support email*	Did not understand	Doubly insecure	—
Mailchimp	*Security email*, *BugCrowd*	No response	Doubly insecure	—
Microsoft	*Security email*, internal bug bounty	Did not understand; later fixed without reporting	Doubly insecure	—
Online.net	*Security email**	Reported as fixed	Secure	18
Paypal	*HackerOne*	Did not understand	Doubly insecure	—
Snapchat	*HackerOne*	Reported as fixed	Doubly insecure	38
Taxact	*Support email*	Did not understand; later fixed without reporting	Doubly insecure	—
Venmo	*Support email*	No response	Doubly insecure	—
WordPress.com	*Support email*	No response	Doubly insecure	—
Yahoo (Verizon Media)	*HackerOne*	Did not understand	Doubly insecure	—
Zoho Mail	*Support email*, security email, internal bug bounty	Closed as non-issue**	Secure	—

the victim website's configuration. Shown below is a sample notification:

To whom it may concern,

This is a vulnerability disclosure arising out of security research at Princeton University. We are computer science researchers affiliated with the Center for Information Technology Policy.

example.com currently offers SMS as an account recovery method. It also offers SMS *as an optional two-factor authentication (2FA) method. It allows users to simultaneously choose SMS for account recovery and 2FA. This means that an attacker who hijacks a user's phone number can take over their account on example.com, without a password compromise. We have attached screenshots that demonstrate this vulnerability.*

We studied the account security measures that control SIM swaps at five major U.S. carriers. We found that all five carriers use insecure authentication challenges that can easily be subverted, allowing attackers to take control of a victim's phone number and intercept their calls and messages.

We also studied 145 websites that offer phone-based authentication and found 17 websites, including example.com, on which user accounts can be compromised based on a SIM swap alone. Currently, in our published dataset, we have redacted your website's name and other identifying information (row XYZ). We plan to release the dataset with all website names in 30 days.

We recommend that you:

- *disable SMS-based account recovery if SMS-based 2FA is enabled.*
- *recommend more secure 2FA options such as authenticator apps to users over SMS.*

Please contact us if you have any questions about our research or recommendations. If you intend to take any actions to improve user account security after learning of our findings, we request that you notify us.

Table A.2 describes all responses we have received at the time of writing (more than 30 days after initial notification). We coded the responses as follows:

- **“Closed as won’t fix”**. The reviewers acknowledged the issue, but decided against mitigation.
- **“Closed as non-issue”**. The reviewers believed the current authentication policy to be adequate.
- **“Did not understand”**. The reviewers did not believe the report was relevant. This includes interpreting our report as customer feedback, and closing our report as out-of-scope.
- **“Fixed without reporting”**. The company mitigated the vulnerability but did not notify us. We discovered the patch during our 60-day re-test.
- **“No response”**. We did not receive any relevant correspondence at the time of writing.
- **“Reported as fixed”**. The reviewers reported to us—at or before the time of writing—that after reviewing our research, the company mitigated the vulnerability.
- **“Template acknowledgement”**. The reviewers acknowledged we had submitted a report on a possible vulnerability in the company’s MFA implementation, but the acknowledgment provided no indication that they had read and understood our report. At the time of writing, we had not received any further correspondence.

B

Appendix to Chapter 4

B.1. Subscriber responses on forums

- https://community.verizonwireless.com/t5/Basic-Phones/How-Do-I-Check-If-Phone-has-Been-used-Before/td-p/333440.
  - (Staff: "We hold off on recycling them [phone numbers] for as long as possible, however depending on the area code and prefix, it can be reused as quickly as 6 months.")
  - ("Cell numbers get recycled (depending on where the number is) rather often.... as in about 6 months. So it is not surprising if he is getting calls from others. ")
- https://old.reddit.com/r/verizon/comments/8d9twz/changing_number_question_does_verizon_recycle/. ("It could be as long as 90 days or as little as a few weeks depending on the carrier, the availability of other numbers in that area, etc.")
- https://www.howardforums.com/showthread.php/1801610-How-Often-Does-VZW-Re-Cycle-Numbers.
  - ("I believe I recall reading that Verizon holds the number for 30 days and then it goes back into the pool to be reissued.")

  - (“I thought it was 60 days before mobile #’s are recycled. I do recall a story a few years ago about a new customer being assigned a phone # previously owned by a high profile person in a big court case in less than 3 weeks in error.”)

- `https://community.t-mobile.com/accounts-services-4/how-do-you-get-your-old-number-back-after-it-s-been-hijacked-8260`. (“More importantly, no one informed me that I had 60 days to get my number back.”)

- `https://www.howardforums.com/showthread.php/1778962-How-many-days-until-cell-number-is-recycled`. (“I’ve heard 90 additional days after the funds expire (180 total I suppose) however your post got me curious and I called [T-Mobile] customer service. She said I would lose the number the day after funds are depleted, which I think she’s in error cause I let it go once way over the 90 days and still had the same number.”)

- `https://forums.att.com/conversations/more-att-prepaid-discussions/recycle-deadline/5defcecebad5f2f606bc34fc`. (“There is a 60 day "grace period" that exists even after your account expiration date.”)

- `https://forums.att.com/conversations/data-messaging-features-internet-tethering/phone-number-recyclingreusing/5ee57d68c17a067c9e6d2dbf`. (“There is no stated policy that is published. The general understanding is disconnected numbers cannot be used for six months. After six months they go into ‘the pool’ and can be reassigned or selected randomly.”)

C

Appendix to Chapter 5

C.1. Visualization of best practices

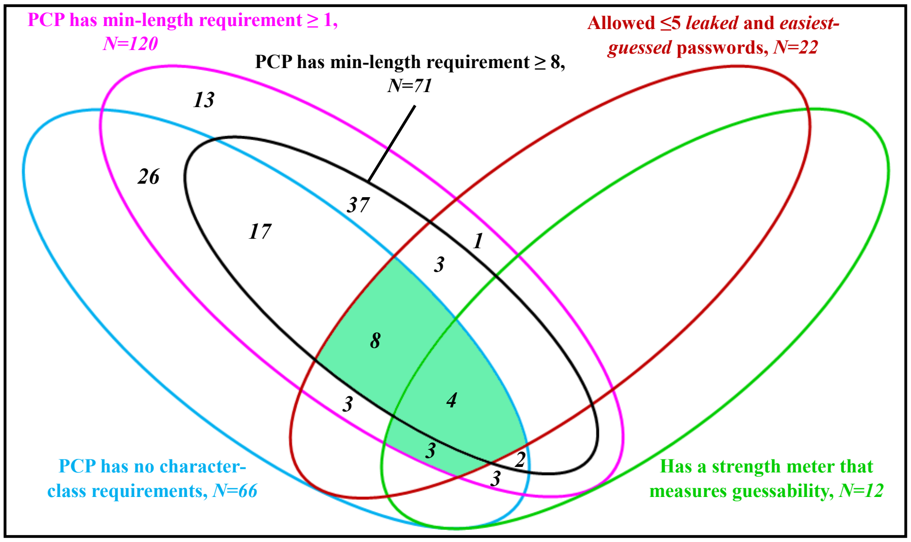

PCP has min-length requirement ≥ 1, *N=120*
Allowed ≤5 *leaked* and *easiest-guessed* passwords, *N=22*
PCP has min-length requirement ≥ 8, *N=71*
13
26
37
17
1
3
8
4
3
3
2
3
PCP has no character-class requirements, *N=66*
Has a strength meter that measures guessability, *N=12*

Figure C.1.: Websites following best practices are in the shaded green area. Unlabeled areas contain 0 websites.

Fig. C.1 shows the breakdown of websites we considered to be following best practices. We considered a website to be following best practices if it allowed 5 or fewer of the 40 most common leaked passwords and easiest-to-guess passwords we tried, required passwords be no shorter than 8 characters, and did not impose any character-class requirements. We also considered websites with a shorter minimum-length requirement as following best practices if they satisfied the other two recommendations and further employed an accurate password strength meter to guide users to choosing strong passwords.

C.2. Lessons learned from attempts at automation

Some readers may wonder why we pursued manual data sourcing methods in this study instead of an automated approach, since doing so may have enabled us to scale up the number of websites tested. As a matter of fact, we initially attempted two automated approaches which we ultimately abandoned due to concerns with completeness and data quality. We include our experiences in this writeup to hopefully serve as useful notes for those who want to extend our work.

We first tried building and using a Selenium-based web crawler to automatically extract PCPs from websites. Our crawler consisted of scripts tasked with parsing and navigating the sites of given domains to find the registration form and the PCP on the form. We leveraged search engine keyword searches to find registration pages (e.g., "join", "create", "signup"), as well as pattern detection of HTML tags and keywords to find and extract the PCP. However, we soon found that it was practically infeasible to develop any general solution; the unstandardized registration flows across websites required us to constantly add code to handle an extremely wide range of UI designs.

Our second approach utilizing MTurk was more successful, but still produced data of dubious quality. We developed and published two separate MTurk Tasks to workers on the marketplace: one to identify registration pages from a given domain, and a second to extract the PCP from a given registration page. For each Task, the Worker—our hired user—was given the domain or registration page, and given a form to input information found such as the minimum-length requirement and character-class requirements. We also included quality assurance questions on the forms to confirm that the Worker had understood the given task and was paying attention. Despite the additional quality assurance measures, we found widespread inconsistency in the clarity of information collected across websites and even at the same website (we made sure to create two Tasks for each website). We concluded that our assurances were not rigorous enough, and that we had also underestimated the difficulty of educating Workers about extracting PCPs.

C.3. Additional findings from § 5.5

1. **Symbol definitions varied among the 37 websites requiring them.** 13 websites counted all 33 symbols we used towards their requirement, and half of the websites counted all but one symbol. The remaining websites below the median counted far fewer symbols, however, including 1 website that counted only #, $, &, and @ (4 symbols), and 2 websites that counted only 9 and 10 symbols, respectively. The most commonly excluded symbol was the space character, which counted at only 15 / 37 websites, followed by the ’, ", and ‘ characters, each counted as symbols at 28 / 37 websites.

   13 websites placed even more restrictions on certain symbols by outright disallowing them in passwords, including the 2 websites that counted only 9 and 10 symbols; any symbol that did not count was not allowed to be in the password.

2. ***1class8* is the most common PCP.** 24 / 120 websites (20%) were using this PCP, followed by *1class6* (22 / 120). *3class8* is the most common character-class PCP (and third-most popular overall), we found it on 17 websites, followed by *4class8* and *DigSym6*, each found being used on 10 websites.

3. **Some websites were using maximum-length requirements that are too short.** 17 websites had a maximum-length requirement below 64 characters—the baseline recommended by NIST—including 1 website with a 14-character maximum length, 3 with a 15-character maximum, and 4 with a 20-character maximum [7]. Setting too short of a maximum length hurts security by preventing users from choosing long passwords that are hard-to-guess.

C.4. Access failure details

Table C.1.: Breakdown of the websites we skipped in our study.

Reason	Websites (N=142)
Inaccessible	69
No registration page	26
No passwords for auth	3
Government website	2
University website	4
Purchase required	7
Never received registration SMS	1
Non-U.S. phone number required	1
Site unreachable from browser	25
Explicit material	6
Non-English	38
Shared reg page w/ already-visited site	29

We tried visiting the top 262 websites on the Tranco list in order to obtain the 120 websites for our study. Table C.1 lists the reasons we skipped the other 142 websites.

C.5. Overall findings for all 120 websites

Table C.2.: PCP, blocklist results, strength meter, and security / usability ratings.

Website	Rank	PCP	Allowed *leaked*	Allowed *easiest-guessed*	Strength meter	Percent weak PWs rejected (security proxy)	Percent strong passwords PWs (usability proxy)
google.com	1	*1class8*	0	0	●	62%	59%
netflix.com	2	*1class6*	20	20		11%	94%
facebook.com	4	*1class6*	8	2	●	11%	94%
microsoft.com	5	*2class8*	14	0		92%	39%
twitter.com	6	*1class8*	12	2		62%	59%
instagram.com	7	*1class6*	8	3		11%	94%
linkedin.com	9	*1class8*	14	0		62%	59%
apple.com	11	*3class8*	4	0	●	99%	7%
wikipedia.org	12	*1class8*	5	1		62%	59%
amazon.com	16	*1class6*	20	20		11%	94%
yahoo.com	17	*1class7*	0	0	●	47%	76%
pinterest.com	21	*1class6*	4	1		11%	94%
adobe.com	22	*3class8*	0	0		99%	8%
vimeo.com	24	*2class8*	18	15		100%	1%
wordpress.com	27	*1class6*	3	1		11%	94%
reddit.com	31	*1class8*	16	4		62%	59%
zoom.us	33	*3class8*	20	20		99%	7%
github.com	34	*1class15 or 2class8*	0	0		92%	36%
amazonaws.com	36	*3class8*	20	20		99%	8%
bit.ly	37	*1class6*	6	3		11%	94%
tumblr.com	43	*1class8*	0	0	●	62%	59%

Continued on next page

Table C.2.: *(Continued)* PCP, blocklist results, strength meter, and security / usability ratings.

Website	Rank	PCP	Allowed *leaked*	Allowed *easiest-guessed*	Strength meter	Percent weak PWs rejected (security proxy)	Percent strong passwords PWs (usability proxy)
vk.com	48	*1class6*	1	1		11%	94%
nytimes.com	49	*1class6*	20	20		11%	94%
flickr.com	51	*1class12*	20	20		100%	9%
dropbox.com	53	*1class6*	20	20	●	11%	94%
soundcloud.com	56	*1class8*	20	20		62%	59%
spotify.com	59	*1class8*	12	6		62%	59%
myshopify.com	60	*1class5*	20	20	●	4%	97%
cnn.com	65	*4class8*	20	20		100%	0%
forbes.com	66	*4class8*	20	20		100%	0%
ebay.com	68	*DigSym6*	11	9		85%	53%
theguardian.com	69	*1class8*	0	0		62%	59%
w3.org	70	*1class8*	0	0	●	62%	59%
paypal.com	72	*DigSym8*	15	5		77%	40%
twitch.tv	73	*1class8*	0	0	●	62%	59%
sourceforge.net	74	*1class10*	0	0		98%	21%
cloudflare.com	75	*2class8*	20	20		100%	1%
archive.org	76	*1class3*	20	20		0%	100%
imdb.com	77	*1class8*	20	20		62%	59%
bbc.co.uk	89	*2class8*	17	14		92%	37%
issuu.com	91	*1class4*	20	20		0%	100%
weebly.com	92	*1class8*	0	0		62%	59%
aliexpress.com	95	*2class6*	20	20	●	84%	57%
washingtonpost.com	96	*1class8*	20	20		62%	59%
stackoverflow.com	98	*2class8*	20	20		92%	37%

Continued on next page

Table C.2.: *(Continued)* PCP, blocklist results, strength meter, and security / usability ratings.

Website	Rank	PCP	Allowed *leaked*	Allowed *easiest-guessed*	Strength meter	Percent weak PWs rejected (security proxy)	Percent strong passwords PWs (usability proxy)
etsy.com	99	*1class6*	20	20	●	11%	94%
reuters.com	103	*4class8*	20	20		100%	0%
tinyurl.com	106	*1class6*	20	20		11%	94%
tiktok.com	108	*3class8*	20	20		100%	1%
wsj.com	109	*2class5*	20	20		85%	53%
wix.com	113	*1class6*	20	20	●	11%	94%
bloomberg.com	114	*1class8*	2	1		62%	59%
sciencedirect.com	118	*4class8*	20	20	●	100%	0%
slideshare.net	120	*1class5*	20	20		4%	97%
imgur.com	121	*DigSym6*	20	20		85%	53%
oracle.com	122	*4class8*	20	20		100%	0%
opera.com	123	*1class8*	0	0		62%	59%
booking.com	125	*3class10*	20	20		100%	3%
indeed.com	126	*1class8*	9	1		62%	59%
businessinsider.com	127	*3class8*	20	20		99%	7%
canva.com	132	*1class8*	2	0	●	62%	59%
godaddy.com	135	*1class9*	20	20	●	94%	30%
cnet.com	140	*DigSym6*	20	20		100%	1%
ibm.com	143	*3class8*	19	20		99%	7%
researchgate.net	144	*1class6*	20	20	●	38%	82%
digicert.com	145	*3class8*	20	20		100%	3%
dailymail.co.uk	148	*1class5*	20	20		5%	96%
slack.com	150	*1class6*	13	4		11%	94%
fandom.com	154	*1class1*	20	20		0%	100%

Continued on next page

Table C.2.: *(Continued)* PCP, blocklist results, strength meter, and security / usability ratings.

Website	Rank	PCP	Allowed *leaked*	Allowed *easiest-guessed*	Strength meter	Percent weak PWs rejected (security proxy)	Percent strong passwords PWs (usability proxy)
nature.com	157	*2class8*	20	20		92%	37%
force.com	159	*2class8*	19	18		92%	37%
cnbc.com	160	*3class8*	20	20		100%	0%
usatoday.com	161	*1class5*	20	20	●	5%	97%
chase.com	163	*2class8*	11	9		92%	34%
walmart.com	164	*3class8*	20	20		99%	7%
hp.com	166	*3class8*	20	20		99%	8%
surveymonkey.com	168	*1class8*	11	2		62%	59%
aol.com	170	*1class7*	0	0	●	47%	76%
yelp.com	171	*1class6*	14	6	●	11%	94%
eventbrite.com	173	*1class8*	20	20	●	62%	59%
telegraph.co.uk	174	*1class8*	20	20		62%	59%
opendns.com	176	*4class8*	20	20		100%	0%
cpanel.net	177	*1class4*	7	7	●	0%	100%
springer.com	186	*DigSym6*	20	20		85%	53%
time.com	187	*3class8*	20	20		100%	0%
npr.org	189	*1class5*	20	20		4%	97%
ted.com	190	*1class8*	20	20		62%	59%
samsung.com	191	*3class8*	19	14		99%	8%
myspace.com	194	*2class8*	20	20		92%	39%
dailymotion.com	196	*3class8*	20	20		100%	1%
themeforest.net	198	*1class8*	0	0		62%	59%
huffingtonpost.com	199	*3class8*	20	20		99%	8%
wired.com	200	*1class6*	20	20		11%	94%

Continued on next page

Table C.2.: *(Continued)* PCP, blocklist results, strength meter, and security / usability ratings.

Website	Rank	PCP	Allowed *leaked*	Allowed *easiest-guessed*	Strength meter	Percent weak PWs rejected (security proxy)	Percent strong passwords PWs (usability proxy)
mailchimp.com	201	*4class8*	20	20		100%	0%
espn.com	202	*DigSym6*	14	10	●	85%	53%
addthis.com	204	*1class6*	20	20		11%	94%
techcrunch.com	205	*1class7*	0	0	●	47%	76%
scribd.com	208	*1class8*	20	20		62%	59%
zillow.com	211	*4class8*	20	20		100%	0%
goodreads.com	212	*1class6*	20	20		11%	94%
unsplash.com	213	*1class6*	20	20		11%	94%
indiatimes.com	214	*DigSym6*	20	20		100%	1%
trello.com	219	*1class8*	20	20		62%	59%
grammarly.com	220	*1class8*	0	0		62%	59%
tripadvisor.com	221	*1class6*	11	3		11%	94%
freepik.com	222	*DigSym6*	0	2		100%	0%
independent.co.uk	225	*DigSym6*	20	20		99%	9%
roblox.com	226	*1class8*	18	6		62%	59%
squarespace.com	230	*1class6*	20	20		11%	94%
foxnews.com	232	*1class6*	20	20		11%	94%
zendesk.com	237	*1class5*	20	20		4%	97%
latimes.com	239	*DigSym6*	20	20		85%	53%
line.me	245	*DigSym6*	20	20		85%	52%
shutterstock.com	246	*1class8*	20	20		62%	59%
livejournal.com	247	*DigSym6*	19	10		99%	9%
wetransfer.com	248	*3class8*	19	4		99%	8%
intuit.com	250	*4class8*	20	20		100%	0%

Continued on next page

Table C.2.: *(Continued)* PCP, blocklist results, strength meter, and security / usability ratings.

Website	Rank	PCP	Allowed *leaked*	Allowed *easiest-guessed*	Strength meter	Percent weak PWs rejected (security proxy)	Percent strong passwords PWs (usability proxy)
intel.com	254	*3class8*	20	20		100%	1%
stackexchange.com	256	*2class8*	20	20		92%	37%
w3schools.com	262	*4class8*	20	20		100%	0%