3

Passwords – A User Perspective

"but his thoughts were so full of the great riches he should possess, that he could not think of the word to make it open, but instead of 'Sesame,' said, 'Open, Barley!' and was much amazed to find that the door remained fast shut. He named several sorts of grain, but still the door would not open, and the more he endeavoured to remember the word 'Simsim,' the more his memory was confounded, and he had as much forgotten it as if he had never heard it mentioned." (Kasim's predicament in Ali Baba and the Forty Thieves) @**TODO** this could be the opening of the chapter / fancychapter

Morris and Thompson were already concerned with user behavior regarding passwords in 1979 [230]. They identified that users choose predictable passwords and that this can be leveraged for attacks. So, they suggested enforcing a certain minimum password length (six characters). At the time, the users were mostly professionals that received training to operate computers and could thus also have been trained to pick less predictable passwords [213]. But as computers were introduced to a larger audience, more people were exposed to password authentication. Naturally, this also induced a growing number of attacks, and it is increasingly difficult for users to defend themselves against them (see Section 2.2). Nowadays, password policies are in place that require not only a minimum of eight characters, but also mandate mixed-case letters, digits and special symbols to start with. The HCI community noticed the users' struggle in the 1990s and that we can – and should – design authentication systems with usability in mind. Perhaps, one of the breaking points where a new school of thought turned up in the literature was a paper by Adams and Sasse in 1999 [3]. The central and novel theme in there was a shift from *fixing* the user to *acknowledging* user behavior and designing for it. The paper managed to see over 1500 citations as of writing this.

This chapter looks at the literature that mostly came after this seminal work. It discusses the users' problems, solutions, feelings, and opinions about using passwords. An essential goal is to give the reader an empathetic perspective and provide background information to understand why it seems hard to come up with viable solutions to make users' lives less frustrating. To get there, we first take a brief look at conducting user research with passwords. Hereafter we disseminate typical coping strategies and solutions. The chapter concludes with a comment on the discourse that has been going on between the very different schools of thought about passwords.

3.1 Methodology: Running Password Studies

Before we report on insights about user behavior regarding passwords, we take a look into running studies that focus on passwords. There are two central aspects that make collecting data particularly challenging: acting ethically and maintaining high ecological validity of the data. In fact, these two goals create an area of tension that demands a critical selection of methods. Komanduri et al. note that "ideally, password studies would be conducted by collecting data on real passwords created by real users of a deployed system" [195]. But this would mean that researchers obtain access to the user accounts that were under investigation. This is ethically questionable [91]. Maybe the researchers themselves are benevolent, but the data is precious and thus could bring attackers to the scene. Since absolute security can barely be guaranteed, it is best to avoid that users disclose their actual credentials during a user study to the researchers.

3.1.1 General Considerations for User Studies as Data Source

If we cannot collect the users' real password in its original form, what is the best way to measure, e.g., cause and effect of novel interventions? There are several alternatives.

Password Creation Tasks

First, one asks participants to create a new password during the user study and stores these passwords as part of the dataset. This approach resolves the issue of real-world password disclosure, but introduces a number of problems. Studies should be ethical and thus transparent for the most part. Hence, the study topic should be known to the participants. However, if participants know that their passwords are studied, this could induce protective reactions to prevent giving hints about their real passwords at all. In that case, the participants' selection strategy does not resemble much to their real-world behavior and thus the ecological validity of the data is low [284]. Although it appears trivial, Fahl et al. suggest that in this scenario, asking the participants whether they had acted like they normally would is a suitable indicator that helps in weighting the data [101]. It is also recommendable to give users a specific scenario that allows them to immerse themselves in the task. Komanduri et al. argue that having participants create passwords for fictional email accounts leads to more authentic behavior [195]. Some users, however, are less protective and provide one of their real passwords regardless of the instruction (e.g. 26.5% in Fahl et al.'s study [101]). The result is the same as if the purpose of the study was concealed through an act of deception, which is occasionally done in psychology studies (for a discussion see [311]). For example, it can suffice to tell participants a convincing *cover story*, e.g. that the purpose of the study is to do a usability test of a social networking site, which also happens to involve an account setup process (see [126]). The data would be ecologically valid because it removes observer-expectancy and other biases. For the researchers, however, it is extremely difficult to tell "real" passwords and "new" passwords apart. Thus, phishing or man in the middle attacks are sometimes carried out. Haque et al. conducted a laboratory study where they told participants a cover story to create new accounts for popular websites [152]. The websites, however, were re-created by the researchers and stored the passwords on their own servers instead of performing actual registrations. Egelman et al. used a proxy server to intercept traffic between the users and a real online portal [98]. They also altered the websites to communicate their cover story that the password had expired. This, however, creates an ethical conundrum, if the dataset is published along with the paper. To allow others to verify that research is valid, reliable, and generalizable, a published dataset is desirable, but in Egelman et al. or Haque et al.'s study this would

Really? or would they choose fake/unused passwords (especially if they don't trust the experimenter?)

especially

Since we recently had the GDPR, another implication is that researchers have to protect this data. ~~Imho~~ Not all research labs have sufficient resources to secure these datasets the same way some companies do.

put real user accounts at risk. Much in the same vein, sharing research about successfully attacking passwords produces a similar dilemma. For instance, one can put forward new cracking approaches (e.g. [217, 232, 271, 351]) that potentially affect common strength metrics (see Section 2.3) – but attackers also benefit from this kind of knowledge. From an HCI perspective, one can also unfold how users select passwords, which allows optimizing cracking efficiency [350, 354].

There are several methods to avoid acting unethically in studies where users are required to create passwords. First of all, studies involving human subjects are assessed by an Institutional Review Board (IRB), especially in the United States. This is done to ensure an ethical study design that is unlikely to cause participants any harm. The IRB might mandate a thorough debriefing of participants and meticulous documentation of the experiment. In Europe, however, password studies are less commonly evaluated by an IRB (or authors fail to mention the process in their papers – often there are remarks that universities do not have such institutions like in [101]). Secondly, one can refrain from releasing the data set, even if user names are removed. In fact, almost all publications on passwords collected during a user study omit publishing the corresponding data set. Only the abstract analysis is published and this is a widely-accepted standard practice, despite the questionable reliability. A rather novel approach that reduces the likelihood of made-up data relies on the idea of publishing differentially private data sets. Here, algorithmically generated noise, which is indistinguishable from the orignal data, is added to the data set to preserve privacy of users. For password frequency lists, passwords could be mangled and extended by generated passwords that resemble real ones. Adversaries lack information whether the data is usable as signal or noise. This way, Blocki et al. managed to release a private frequency list of passwords at Yahoo that Bonneau had already anonymously analyzed in [31]. Moreover, instead of collecting newly generated passwords in plain text, it is possible to store a hashed version. For instance, Wash et al. had participants install a browser extension that logged all form submits that included a password field [348]. To study reused passwords, they hashed and sent them over a secure connection to their servers. As long as a slow hash function and a strong salt are used, this approach is uncritical. However, it merely allows observing if the hashes match on several sites. Finally, as a last option to collect password data, researchers can log meta-data instead of passwords. For instance, Von Zezschwitz et al. used a "meta password" that described the participants' actual passwords [339], but which was insufficient to reconstruct the original. This can include the number of characters, upper-/lowercase letters, digits, and even proactive strength estimations. To collect the data, participants in von Zezschwitz et al.'s study were provided with an offline password analysis tool. They entered their password into that and copy-pasted the result of the analysis into the questionnaire form. If one does not want to examine the passwords qualities, this approach is absolutely feasible. Florêncio and Herley used a similar approach for the large-scale data collection with around 500000 participants to avoid running into privacy issues [109]. The information transmitted to the logging server was pseudonymized and contained only meta features. The only downside is that one cannot run further analyses on the passwords, e.g. if a new strength metric is considered.

Retrospective Self Report

If one wants to refrain from having users create a new password, one can study their past behavior in different ways. In its simplest form, participants are simply asked to describe how they create passwords. Stobert & Biddle did this in extensively to create the Password Life Cycle model [303] (see Section 3.2). Ur et al. used interviews to find out what users do to make their passwords stronger [330]. Das et al. found out through retrospective interviews that social contacts have a strong impact on users' security decisions [68]. Most commonly, however, typical online surveys feature a number of questions about personal behavior and attitudes, e.g. [4, 130, 201, 259, 285]. Questions about passwords are easy to implement in a survey and respondents can always choose how much they want

to share. However, one has to stay aware that social desirability lowers the reliability of the data: since the media also to their part in shaming users for picking "bad" passwords, people may respond dishonestly about their password behavior. Many people are uncomfortable admitting their password is as simple as 12345. Another problem results from fading memories. Since most users have more than one password, it might be difficult for them to recall the correct past behavior. However, not only password studies suffer from this bias, but any study involving self report in general.

Principles

From user research in USEC of the past decades, Krol et al. derive a set of general principles that researchers ought to consider when conducting experiments in security and privacy [200]. We can integrate password studies in there:

Primary Task Creating a password should not be the sole task in the study. Instead, participants should achieve a primary task by authenticating with passwords, e.g. using a new system for a period of time [40]. The reasoning behind this principle is that security tasks are secondary tasks and this constraint needs to be reflected in the experiment. However, re-focusing on a separate primary task is not always possible, e.g. in surveys.

Realistic Risk Users should be able to realistically estimate the risk for secure interactions. As mentioned above, Komanduri et al. suggest carefully selecting real-world scenarios to achieve this [195].

No Priming Whenever human behavior is studied, experiments should avoid influencing and biasing participants with certain information. This avoids unnatural behavior.

Double Blind Experiments If possible, the person carrying out the actual experiment should be involved in the planning and design of the study. Moreover, the participants should not know the details of the study, either. On the experimenter side, unconscious bias and influence is mitigated, and participants also do not know the "treatment" they receive (if any). Although this is a desirable goal, there is little evidence that experiments are usually carried out in this way.

Context Definition To increase internal validity, it is necessary to define the terms *threat model*, *security*, *privacy*, *usability*, depending on which are relevant for the study. A precise definition avoids misunderstanding and improves transparency, credibility, and trustworthiness of the experiment.

These principles can serve as a rough quality assessment of presented research, although in many instances, not all principles will be fully addressed.

3.1.2 Analyzing Password Leaks and (Semi-)Public Data

Instead of users creating new passwords, it is possible to make inferences about their behavior from already existing data (for an overview see Table 3.1). Password frequency lists are readily available on the Internet [1]. The sources can often be traced back to illegal attacks, which makes the use of such data somewhat questionable. However, it is a widely accepted method to contrast real-world and study behavior. The data set that has probably been studied the most originates from a breach at RockYou, a software development firm specialized at games for social networks. In 2009, an attacker used an SQL injection to download around 32 Million plain-text passwords. For instance, Veras et al. visualized semantic properties of passwords in this dataset and highlight the high occurrence of dates

[1] An example repository containing a wide range of leaked passwords is available under `https://github.com/danielmiessler/SecLists/tree/master/Passwords` *(last accessed 09.01.2018)*

in there [334]. Wheeler relied on it to build the zxcvbn password strength estimation system. More often, though, these data sets serve as training data for password guessability benchmarks. Weir et al. took the RockYou passwords to train their PCFG which served as a demonstration that entropy is not a feasible strength metric for user-chosen passwords. Afterwards, it was integrated into the training set of PGS. PGS is mostly used to gauge passwords collected through a user study, e.g. under different policies [282] or interventions [326]. In conclusion, publicly leaked data sets can often serve as ground truth for studies that aim to provide new insights.

Table 3.1: Example password leaks of the past five years (data source: `https://haveibeenpwned.com/PwnedWebsites`). Some of the data served security researchers to analyze user behavior and create more effective strength estimation algorithms. *multiple leaks from different years

Data Source	# PWs	Year	Usage Examples
MySpace	360 M	2008*	[72, 67, 80, 204, 221, 282, 328, 333, 354, 351]
RockYou	32 M	2009	[15, 25, 29, 36, 72, 81, 189, 98, 175, 195, 227, 282, 333, 346, 350]
Dropbox	68 M	2012	[26, 126, 241]
Yahoo	0.5 M	2012*	[26, 67, 143, 168, 189, 221, 282, 331, 344, 354]
LinkedIn	164 M	2016*	[72, 113, 173, 193, 208, 333]

3.1.3 User Study Methods in Password Research

Research in USEC takes advantage of the toolbelt of HCI research in general. In the following, the most common methods for password studies are portrayed.

Laboratory studies To gain the greatest control over experimental parameters, laboratory studies are the go-to method. Both qualitative and quantitative studies are carried out in the lab, with a slight surplus of qualitative studies. Among the most common methods, one finds (semi-)structured interviews ([4, 130, 303, 330, 353]) and usability testing ([98, 120, 122, 140, 169, 223, 225, 262, 366, 340]). Password memorability can be studied in the lab ([98, 101, 119, 225, 364]). Studying short-term recall usually follows a mental-rotation task (e.g. [198, 366]). Long-term memorability studies in the lab are less common, because they require participants to return to the lab, which may be too bothersome for many. For alternative authentication schemes, this may, however, be the only option.

Although technically it is not a "lab" environment, café studies sometimes are closely related to lab studies. Von Zezschwitz et al. collected qualitative and quantitative data on participants' past password behavior by inviting customers in a café to join them for a free coffee [339]. The experimenter has almost as much control as in a lab to answer certain research questions. Only the surroundings may distract somewhat. Other methods like participatory design / co-creation ([61, 253]) and focus groups ([89, 149, 153, 294]) are possible, but less frequently reported than interviews and usability tests. A common drawback of lab studies lies in the high costs, time consumption to carry them out, smaller sample sizes, and reduced ecological validity.

Field-Studies Field studies for password research come in many flavors. **Online surveys** are among the most common methods used in the field ([129, 146, 152, 167, 201, 221, 259, 336]), due to their lower cost and comparatively easy implementation. Other advantages like increased sample size and more diversity in the data speak in favor of online surveys. Survey tools like surveymonkey.com come in handy, but usually lack seamless integration of interactive prototypes. If a prototype should

be evaluated through a survey, however, one has to either implement the entire survey structure or redirect participants from the survey platform to the prototype and back. Surveys are also the weapon of choice if there is an opportune moment that is worth studying. Mazurek et al. took the opportunity to distribute online questionnaires after a new password policy was introduced at CMU [221]. Fahl et al. profited from a similar situation at Leibniz-University Hannover, and Renaud et al. could even distribute surveys on the same topic across multiple years in this way [257]. Interestingly, there does not seem to be a special, standardized survey construct to measure the usability, respectively user experience, of password systems. Other Human-Computer Interaction (HCI) sub-fields more frequently use, for example, the NASA-TLX, Positive Affect and Negative Affect Scale (PANAS) or AttrakDiff constructs to establish comparability with other studies. Notable exceptions were reported by Kraus et al., who used AttrakDiff to evaluate emoji-based authentication [198]. The NASA-TLX was used by Fraune et al., [125], Sherman et al. [289], and Yang et al. [366]. Lately, the Security Behavior Intentions Scale (SeBIS) gains more attention, because it serves as a self-assessment that can help the interpretation of actions taken during a user study [96, 93, 348, 349].

A special kind of online studies that has been extensively used and propagated by CMU researchers leverage **crowd-sourcing** platforms like the Amazon Mechanical Turk (mTurk)[2]. Survey respondents are recruited by paying each one a small amount of money for a valid response. This way, increasing the sample size is straight-forward, if many users have already signed up on the platform and are eligible for the Human Intelligence Task (HIT). Workers (known as "turkers") form an increasingly diverse population [261], which is another benefit. In password research, for instance, Kelley et al.'s high-impact work on password guessability collected around 12000 passwords using mTurk [188]. Ur et al. had participants rate the strength and memorability of a given set of passwords, which allowed them to identify certain misconceptions [327]. Mazurek et al. compared features of passwords created by turkers to real passwords of students and staff at their university [221]. They take the large similarities of the two data sets as evidence that passwords created during an mTurk study are a reliable and valid data source, so there is no urgent need to analyze passwords of a deployed system. Shay wrote a PhD thesis specifically about evaluating password policies with crowd-sourced data [280]. In many cases, e.g. [286, 282, 326], the primary task is to create a fictional account or merely a password under certain constraints, which apparently violates Krol et al.'s study principles [200]. It is especially interesting that most studies are announced as some kind of password study, probably mandated by IRBs . But Mazurek et al.'s work demonstrates that this limitation is bearable. Moreover, studying the long-term memorability of passwords is facilitated, because participants can be invited to return through an internal, anonymous messaging system. It is also possible to create more complex study designs with mTurk, e.g. if multiple device types should be used by the turkers to create passwords [226]. Despite the wide range of advantages, there are shortcomings of crowd-sourced approaches as with any study method. First, turkers are incentivized to complete as many HITs as possible on the platform to earn money. Thus, completing a survey by providing quick answers without reading the questions could lead to unreliable data. Instructional Manipulation Checks (IMCs) and attention check questions (ACQs) can mitigate this problem [240, 244]. Turkers are only paid if the commissioner accepts the HIT as valid, thus IMCs and attention checks are useful indicators here. Moreover, as of now, the mTurk platform can only be used in certain countries, e.g. the USA or UK. European alternatives exist, but are not yet par in terms of user base, response times, and feature set [244].

Aside from surveys, **diary studies** about passwords have proven feasible in the past. Inglesant and Sasse found out through a diary study that employees struggle with frequent password changes, which might not have become evident using other study means [170]. Hayashi and Hong used this method

[2] `https://www.mturk.com/` *(last accessed 10.01.2018)*

to analyze password re-use across different computers, services, and organizations [157]. Since password authentication is a secondary task, keeping a diary of authentication events helps participants provide reliable behavioral data. However, it requires much effort to continuously stay aware of one's actions to log them. To avoid that participants forget logging and other self-reporting bias [349], it can be worthwhile to ask them small questions in situ. This method, known as **Experience Sampling Method (ESM)**, requests short responses either in predefined intervals or when the system detects a relevant event. ESM has not seen much attention in password studies on the web (Lyastani et al. provide one exception [212]), but mobile authentication has been studied with this method [153]. Users carry their personal mobile with them almost all the time, so there is a high chance of successfully receiving the experience sample. ESM was also found useful for studies about security warnings in browsers [8, 104].

Perhaps, ESM is underused for password studies, because it is possible to automatically detect such events and survey the participants before and/or after the **automatic data collection**. Florêncio and Herley conducted one of the largest studies to date on password habits with this method [109]. Their intention was to find out among other things A) how often people type passwords, B) how many sites share a password C) how many distinct passwords a user has, and D) how the strong the passwords are. Working at Microsoft, they were granted to utilize the Windows Live Toolbar for Internet Explorer to collect in-the-wild data from up to 500,000 users during three months of running the collection. They conceived the method of protected password lists (PPL) to avoid intruding into people's privacy – a kind of meta description of passwords which is sent to the logging server instead of the original password. It was thus not possible to trace the incoming data back to a specific user. However, there are a number of limitations this method. The authors point out that the anonymity of the incoming data-stream might have resulted in over-counting of entries. Also, it was not measured how long the actual password entries takes. If users only used regular dictionary words without any modification as their passwords, the key logging module of the toolbar would have recorded a password reuse event (PRE) every time the user entered that word – also in regular online communication. Nevertheless, the fact that users are typically focused on their primary tasks, background logging helps to collect unbiased data with high ecological validity.

A final option to study passwords in the field is to collect and analyze them in an already deployed system (**in-situ evaluation**), which would be the ideal data source, according to Komanduri et al. [195]. Brostoff and Sasse utilized a coursework system to evaluate Passfaces as alternative to passwords [40]. Similarly, Renaud et al. used a coursework tool to evaluate the effectiveness of different password nudges [257]. Mazurek et al. gained access to passwords of their University's Single-Sign On (SSO) and were able to break down differences in password selection behavior by departments. As one of the few exceptions from the industry, Bonneau analyzed a private password data set at Yahoo [31] and Amazon [36]. The data is highly ecologically valid and diverse if it originates from a real product or service. However, if interventions are implemented as part of a study, this might have negative consequences for both the service provider and the users. For instance, in an A/B setting one intervention to influence password selection might in fact lead to weaker passwords and put users at risk. Each user is a critical potential source of revenue for service providers, so tampering with the sign-up procedure might lead to higher bounce rates and consequently financial loss. High stakes like this make it difficult for researchers to convince stakeholders to cooperate on a study. In conclusion, it is unsurprising to see only rare instances of password studies carried out with deployed systems in public environments, although the insights gained might be invaluable.

3.1.4 The Bottom Line: Emerging good practices and tools

Using one of the study methods above is the first step to get closer to answering the research methods. However, to get the full picture of the studied phenomenon, a **triangulated** approach appears to be the only option. For example, Wash et al. combined a survey with log analyses to study password reuse and self-report issues [348]. A multi-tiered approach like in Von Zezschwitz et al.'s study helps to identify themes first (formative stage) and quantify them later (summative stage) [340]. Similarly, Huh et al. were able to refine their concept of system-initiate user-replaceable passwords through triangulation [168]. Adams and Sasse conducted qualitative interviews to follow up web survey results [3]. If constraints allow for only one method, it is recommendable to consider how to collect both quantitative and qualitative data points. For instance, in online surveys that evaluate a novel password intervention, it is always feasible to collect quantitative metrics (e.g. usability and password strength) and qualitative data (reasoning, explanations, feedback) to put study results into context [3]. Those eager to find a starting point for Usable Security and Privacy (USEC) experiments probably find essential aspects in Krol et al.'s principles for experimental design [200]. The methods described above can be drawn on to fulfill those principles, which is aimed at in Part II and III of this thesis.

3.2 Password Coping Strategies and Risky Behaviors

Passwords are the cornerstone of *knowledge-based* authentication. And although "knowledge" can be stored inside and retrieved from computers, it is still a human capability to learn things and hereafter "know" them. So, humans are a large factor in the equation of knowledge-based authentication. Their actions and behavior to gain knowledge on passwords deserve to be studied in detail.

Some cybersecurity researchers started blaming system failures and vulnerabilities on users. For instance, Feldmeier et al. stated in 1990: "The main weakness in any password system is that users often choose easily guessable passwords: English words, names, trivial extensions to English words, etc., because they are easy to remember" [102]. It quickly became a dictum that users were the "weakest link" in the figurative authentication chain [266]. However, since the late 1990s, the HCI advocates that systems take into account user capabilities and not the other way around [267]. Adams and Sasse postulated in 1999 that service providers acknowledge that "users are not the enemy", which is one of the most influential position papers on the topic [3]. In that paper, they provide four central challenges in password authentication that users face: 1) Users have to deal with multiple passwords, 2) users do not intuitively create strong passwords 3) password procedures and work practices might conflict and 4) users have a sub-par understanding of organizational security issues. Those challenges are often too hard to come by in everyday password authentication [86]. As a consequence, users develop coping strategies to reduce their task load. This early framework has since been fed with numerous research studies and is still valid today.

Stobert and Biddle formalized user challenges and behavior in the "Password Life Cycle" (see Figure 3.1), which they arrived at through qualitative interviews and coding the participants' responses [303]. It starts out with the challenge to **choose a password**. Coping strategies at this point revolve around reducing effort, e.g. to memorize the password. Including personal or personally meaningful information comes natural to users. Others include pointers to the time they created it, or word-associations about the website-content. Mnemonics are found with some users, especially those aiming to secure their account. In essence, however, users often memorize their *coping strategy*, to recall their password. Consistent strategies reduce the effort effectively. Even if complex policies mandate a change to the first-choice password, users have a go-to strategy to deal with this situation,

e.g. by appending a preferred symbol. When people create passwords, the most common action is to **reuse an old password**. This is not always possible, so users need to maintain and **commit to** a number of passwords. Hayashi et al. observed in a diary study that users categorize accounts [157] in different ways. There is also a mix of different password retrieval methods at the commitment stage. A survey in 2017 from Pew Research Center with N=926 participants found that the vast majority commits to their passwords by memorizing them in their heads (preferred strategy for 65% of the respondents) or by noting down the password (49% do this, and it is the preferred strategy for 18%) [239]. Some respondents also either saved passwords in their browser (18%) or used a dedicated password manager (12%), but this appears to be a negligible go-to strategy (5% of the respondents). Once the user has committed to a password, they **live with it**, even if it produces difficulties in certain situations. For instance, the question "when is it time to change the password?" falls into that stage of the cycle. Finally, if passwords are not actively used on a regular basis, or were recently changed, it is very foreseeable that users **forget their passwords**. The password reset mechanism helps users cope in this situation. The two options at this point are to either create a completely new password (which increases the likelihood of forgetting it), or to reuse one (which potentially reduces the number of unique passwords). Then the cycle starts over. In the following we shed light on the actions and consequences at the different stages.

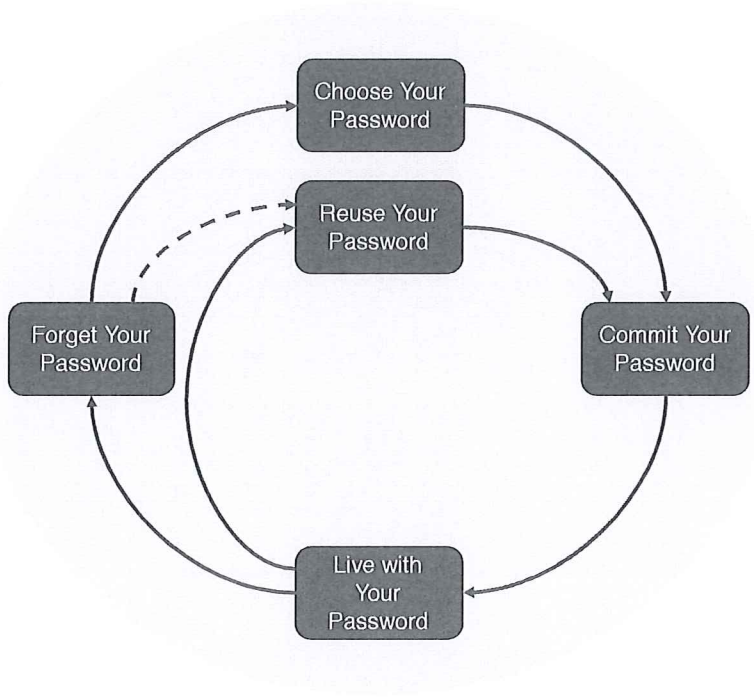

Figure 3.1: Stobert and Biddle's "Password Life Cycle" [303] models typical stages of password behavior.

3.2.1 Weak Passwords

Why do users select weak passwords? First of all, selecting a strong password is hard for most people. Picking up the definition of a strong password ("something that is easy to remember, but difficult to guess" [24]), users do not struggle with the first part of the sentence, but the latter. Users do not intuitively know what makes a password *difficult to guess* [175], and not even security researchers reach ultimate consensus on that matter.

This also sounds informal

Let us look at the first part: something that is easy to remember. Numerous studies have looked at what people do to make their passwords easy to remember. For instance, personal information is easy to remember ("TobiasSeitz"), as is that of close ones ("LenaSeitz") and pets ("Fonsi&Alois") [41, 204]. Veras et al. found that dates are very commonly found in passwords [334]. Looking at the top 25 most-used passwords[3], a list published after each public data leak, we can easily spot more patterns. One group consists of "keyboard patterns" (qwerty, qazwsx) and "number sequences" (12345, 123456, 1234567489, 1234567, 123123). The remainder fall into the "likings" (football, monkey, iloveyou, starwars, dragon) and "password thematic" (Password, letmein, admin, welcome, login, passw0rd, master, hello, trustno1) categories. All of these passwords are particularly easy to remember, which was quantified by Chiasson et al. [51], but extremely predictable. However, they are often still allowed at many websites [276]. Interestingly, such password lists differ marginally across countries [336, 344]. Consequently, users' **desire to create memorable passwords** naturally leads to more obvious selections.

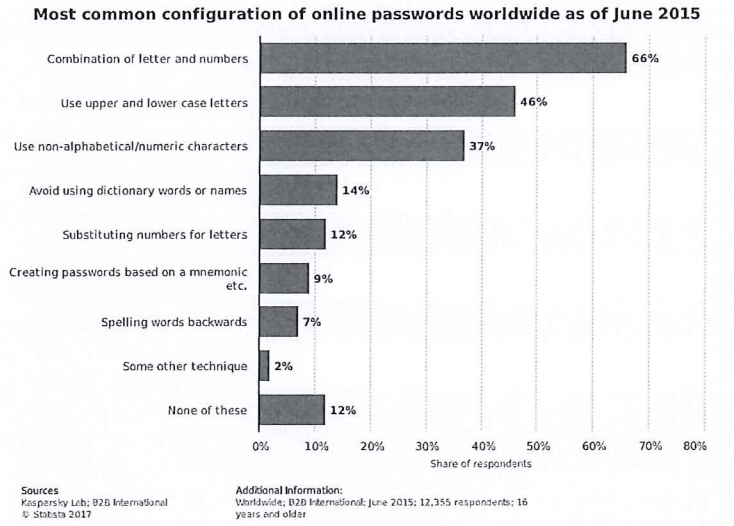

Figure 3.2: @TODO replace with own figure built from Kaspersky Lab report data Survey results - how do people select their passwords (self report)

Another reason for the prevalence of weak passwords lies in incorrect *mental models*. Mental models are descriptions of how humans make sense of functions and system states [337]. Some users who aim to create stronger passwords still fail because their strategies to accomplish the task are predictable. Gaw et al. found that mangling is predictable [129]. Ur et al. pointed out that users understand a great deal of password security, but their modification behavior of existing passwords is sub-par [330]. Moreover, they provide evidence that users succeed to identify strong passwords, but only certain characteristics are misleading [327]. For instance, including digits yielded significantly higher subjective strength ratings, but does not always effectively improve strength. What is more is that participants in their study showed a skewed understanding of password attacks. For instance, 34% of respondents thought that a strong password needs to withstand at least 50 guesses by an attacker. Therefore, trying to fend off a dictionary attack might be futile if users do not know that attackers attempt to crack the password trillions of times. Ur et al. conclude that feedback should thus inform users about attack scenarios which helps them assess the risk more realistically. Although

[3] SplashData publishes such a list each year, for 2017: `http://fortune.com/2017/12/19/the-25-most-used-hackable-passwords-2017-star-wars-freedom/` *(last accessed 12.01.2018)*

When reading this sentence, I was curious which paper reported that. But it was difficult to know which paper to look into [330] or [327]?

it is sometimes argued that passphrases, i.e. a combination of multiple dictionary words, are often as secure as passwords from a richer character set [284], users fail to create strong passphrases, too. Bonneau studied the linguistic properties of passphrases at Amazon and noted that an attacker could easily model user behavior to effectively crack passphrases [36]. So, in other words, the mental model "passphrase = secure" is also wrong and problematic. Finally, mental models also play a role for risk assessment in organizations. In many cases, employees underestimate the security threat that companies face day to day, which leads to insecure password practices [3, 353]. To summarize this point, users fail to create strong passwords, because in many cases **erroneous mental models** stand in the way.

What the media often call "laziness" and "stupidity" [337], may in fact be the rational rejection of password security. Riley pointed out that users are well aware of "better" behavior, but they often ignore it by choice [259]. Florêncio et al. argue that this behavior is absolutely rational, because some accounts do not require strong protection and it would be impossible for users to follow all security advice given by experts [114]. A case example of an account that may be valuable for some, but not for others is LinkedIn. The social-media platform which is focused on business connections suffered a severe data breach in 2012, but only became aware of a much larger leakage in 2016. Many hashed passwords had leaked and thus the affected users were prompted to reset their passwords. Huh et al. investigated the reasons (not) to reset the password and in many instances, people said that they do not use the service very often, did not want to, or were not really concerned about the risks [167]. Ur et al.'s results also indicate that users do know in theory what makes a strong password [327]. A survey commissioned by LastPass reported that one group of users often do not care about their accounts if they are not meaningful to them [203], while the other group is overly careful, but there does not appear to be a "just fine" area of behavior. The first group may believe that stronger passwords do not accomplish protection anyhow [112]. Herley and Pieters argue that it is also difficult to objectively falsify claims about security precautions, which would help to debunk unjustified security advice [162]. So, in essence, users choose weak passwords because they think **the account is unimportant** and therefore it is fine to ignore security advice.

Lastly, the literature lists a few other reasons that lead to weak passwords. Groß et al. looked at the association between cognitive depletion and selection behavior [141]. They reach the conclusion that if cognitively challenging tasks precede password selection, the resulting passwords tend to get weaker. Von Zezschwitz et al. point out that a user's first passwords are created probably in teenage years when security precautions might be much less evident to people [339]. For important accounts, these early passwords are modified, but still persist for many years after they were first committed to. Mobile phones are another factor that steadily gains more importance. Under lab conditions, Yang et al. found that participants included more lowercase letters in passwords, if they created them on a smartphone [366]. Von Zezschwitz et al. corroborate the findings and argue that passwords created on smartphones are much less diverse than their desktop-counterparts, because they are shorter and contain mostly lowercase letters [340]. Two years later, Melicher et al. studied the interoperability of passwords on different platforms [226]. Interestingly, the passwords created on mobiles were only marginally weaker than those created on a desktop, but had more potential to lead to user frustration, especially if requirements were too bothersome. The individual keyboard on a mobile influences frustration levels, too [151]. In summary, contextual factors like cognitive depletion and the device used during password creation have a notable impact on password strength.

We thus identify four main themes from related work that cause weak passwords: 1) Weak passwords are more memorable. 2) Hard-to-change mental models prevent the creation of stronger passwords. 3) In many situations, users rationally reject the effort to create a strong password. 4) Contextual factors notably influence selection strategies. In the following, we dissect another coping strategy, which is potentially even riskier than selecting weak passwords: password reuse.

3.2.2 Password Reuse

The primary reason for password reuse is the mere fact that users create new accounts on a regular basis, and it is logical to make sure to be able to log in later by choosing a secret they already know. "Password overload" essentially frames the problem [365]. "Memory interference" postulates reuse as a coping strategy [51]. In 2007, Florêncio and Herley conducted a large-scale study that empirically showed the challenges and coping strategies regarding this overload [109]: Users of the Windows Live Toolbar had 6.5 distinct passwords, each of which was used for 3.9 different websites. During the data collection period, users logged into 25 accounts on a regular basis, and typed around 8 passwords per day. Although keeping track of the multitude of username-password combinations is a tough challenge, most users still rely on their memory instead of other tools [239]. Users realize the challenge is hard, but Woods et al. argue that users underestimate their capabilities when it comes to memorizing passwords [361]. Consequently, many people do not try to create a richer portfolio of passwords than the numbers from the 2007 study showed. More recent numbers show that between 39% [239] and 76% [64] of users rely on reuse as coping strategy. Since social desirability bias could lead to dishonest survey responses, the "dark figure" might even be higher, because, as Inglesant and Sasse put it, "*Users see 'good' passwords (that are memorable and conform to the policy) as a 'resource', which they continue to use for new applications even if the original use is no longer allowed.*" ([170]). What is more is that the user-name is part of the authentication process, and users pick different aliases, pseudonyms, emails for different accounts [157].

All this is consequential for the overall online security of an individual. The more a user relies on reuse passwords, the more severe phishing attacks and data breaches become. The metaphorical "Domino Effect" describes the situation after a breach: When an attacker obtains a password form one user's account, all the other accounts might fall with it [171]. It's enough to know even only one low-value password to crack a large part of high-value passwords via predictable mangling rules [67, 152]. Still, password reuse is difficult to mitigate, so Ives argues to look into understanding the specific approaches better [171].

One such approach is to categorize passwords by different criteria. Here, we can borrow the "Mental Accounting" theory from Behavioral Economics to describe users' risk assessments. [307, 315]. Users put their passwords into mental accounts that help them recall them later. Wash et al. argues that frequently entered credentials are reused more likely than seldom used ones [348]. However, interviewees in Stobert and Biddle's study reported the opposite behavior [303]. Florêncio and Herley noticed that strong passwords (in terms of entropy) are less frequently reused than weak passwords [109]. Here again, Wash et al. observed the opposite: participants in their study prioritized stronger passwords as reuse candidates [348]. If the latter is true, than this is another indicator for problematic mental models regarding risk assessment: users try to follow security advice and prefer strength over uniquenessm which is typically also advised. Nonetheless, strategies (or at least results) appear to fluctuate throughout the years. Many users cluster their secrets regarding the usage purpose [152], e.g. banking, social media, communication, shopping, etc. [238, 302]. Bailey et al., however, argue that users do not really respect the usage purpose, but the value, importance, and meaningfulness of the account [15]. In that sense, users do not appear to care if the password ought to protect financial information if the website does not mean much to them. Radke et al. found users in their diary study tended to create unique passwords for important accounts, and reuse passwords for less important ones [251]. Still, probably all users have a "go-to password" that is tried first for an account whose value is uncertain in the beginning [303]. If password requirements disallow the preferred choice, the go-to password might not work anymore. Typically, users either pick another password from their portfolio in that situation, or they mangle their first choice until all requirements are met. As we show in Chapter **??**, neither strategy is necessary if the go-to-password shows certain features. But in case the policy mandates a change, Gaw and Felten have laid out that user-chosen mangling

strategies are predictable, too [130]. Besides, policies might not be the only reason for password modifications. Over time, users sometimes are exposed to new security advice or other realizations that their previous password strategy is generally considered weak [339]. Taking the current (already memorized) credentials, and applying the recommendations to them is an obvious choice.

Florêncio et al. describe reuse, categorization, memorable passwords, and mangling as "finite effort" [114]. They meticulously lay out that password these strategies are not only common, but inevitable and necessary. Even experts in cybersecurity show similar patterns [211, 304]. The major difference between mainstream users and experts is that the latter are more articulate and considerate about their coping strategies.

In conclusion, we can hang on to the idea that password reuse is a necessary coping strategy. Zhang-Kennedy et al. motivate that it is not even "bad" or "risky" per se [373]. The challenge is to do it "right". However, motivating users to alter this particular aspect is comparable to motivating a smoker to quit their guilty pleasure: Abandoning reuse does not show a visible immediate payoff (nor does quitting smoking). In many cases, it does not cause harm, even though 16% of US-Americans have experiences someone else taking over their accounts [239] (a similar percentage of active smokers develop lung cancer[4]).

3.2.3 External Storage

To avoid memory interference, many users resort to writing down their passwords. In its simplest form, a sheet of paper that holds user name and password suffices. Roughly half the users reportedly do this [239]. Many users also use a dedicated note-book that keeps their passwords in one place [197]. Interestingly, some manufacturers offer "password-logbooks" to help users organize their credentials. Kothari et al. collected customer reviews about the ten most reviewed logbooks and analyzed their content to derive a mental model of password security [197]. They were surprised how many people apparently use one of those logbooks and what their motivations are. For instance, customers often loved the inconspicuousness of the books and gave them away as presents. Age-related memorability challenges were also a central theme. Many people acknowledged the security risks but were unsure if a piece of software would be more secure than the book. Password logbooks can become a single point-of-failure. Digital files on the computer are often almost as accessible to local attackers (friends / spouses). In the workplace, writing passwords down on sticky notes [59] leaves the credentials wide open to anyone passing by and enforcing different behavior is difficult. Nevertheless, Herley and Van Oorschot generally advocate writing down as coping strategy, as long as the notes are stored in a fairly secure location [163].

Using a password manager (PWM) is a more sophisticated way of "externalizing" passwords. In essence, a password manager is a digital representation of the "logbooks" described above, but comes with many helpful extra features, like easy access, encryption with a master password etc. A plethora of services and tools exist in different flavors (e.g. built into browsers, third party programs, browser extensions, free vs. premium, cloud-based vs. local). Notable representatives include LastPass (freemium/subscription), 1Password (premium/subscription), KeePassX (free, one-off payment for apps), and Dashlane (freemium/subscription). Arias-Cabarcos et al. evaluated popular PWMs and suggest that Dashlane provides the the most feasible usability/security trade-off [11]. However, surveys have often revealed the low adoption rate of password managers [239], mostly because users feel secure enough with their current management habits, or due to financial hurdles and distrust [64, 100].

Lyastani et al. recently investigated the impact of using a PWM on password strength and reuse [212]. Those PWMs which included a generator had a positive effect on overall password strength and

[4] https://www.verywell.com/what-percentage-of-smokers-get-lung-cancer-2248868 *(last accessed 13.01.2018)*

diversity in the large sample studied in-situ. But existing user strategies thwarted a boost in security, e.g. if a built-in PWM is solely used to store reused passwords. Users often benefit from automatically filled login-forms and do not have to type their passwords anymore, which is a huge usability plus. Autofill moreover mitigates most phishing attacks, because the PWM verifies the domain. As with analog notepads, PWMs are a single point of failure, in case the master password is weak. This is one of the few instances in which a strong password is recommended without restrictions. Password managers constitute a honey pot for attackers, who exploit security vulnerabilities of the software to gain access to user passwords [33]. The situation is aggravated if passwords are synced to the manufacturer's cloud storage, although this is recommended by Yee to enable seamless availability [368]. To summarize, password managers are a feasible solution for many users, but adoption rates are still fairly low.

3.2.4 Fallback Methods

Coping with a forgotten password shows more particularities of user behavior. If offered by service providers, the easiest way to handle the situation is to obtain a password reset link via email and create a new password. In the past, password resets had been responsible for a large portion of helpdesk calls [266], but self-service procedures have successfully mitigated the issue. Users tend to reuse passwords or mangle an old password when they forget and reset their credentials [303]. In some cases, however, password reset links are replaced by personal knowledge questions [32]. These often present a great risk to user accounts because of the statistical probability involved (e.g. "what was the make of your first car" has a predictable distribution), or the information is findable on social networks (e.g. "what is your city of birth"). Consequently, a strong primary password can be dominated by a weak fallback question and voids user efforts to secure their account. Perhaps the only strategy to maximize security within this scheme is to provide bogus answers to secret questions to fend-off statistical attacks [5].

3.2.5 Account Sharing

Users often share their accounts, e.g. with close-ones, relatives, or co-workers [185, 285]. Security advice generally discourages sharing passwords because it increases the likelihood of leaked credentials. However, Singh et al. point out that this behavior is absolutely intentional, and does not originate from lack of understanding the risks involved [294]. Password managers also integrate sharing features to give others quick and easy access to a set of passwords [205]. Account sharing plays a role in the overall security strategy, but personal passwords are usually less influenced by sharing considerations.

3.2.6 Summary

We can identify themes in coping strategies at each stage of the Password Life Cycle [303]. Coping is a natural reaction to "impossible demands" arising from Password overload [267]. All strategies can be justified from an economic point of view [114], although each of them generates security risks of varying severity. Password reuse is arguably the most severe problem, followed by selecting too obvious and weak passwords. It is critical to be consistent with one's own reuse strategy and pick strong passwords in a select number of cases, e.g. master passwords for PWMs or central hubs like email accounts. In the following, we discuss how research has tried to influence risky behavior and support users in safe behaviors.

[5] `https://lifehacker.com/use-fake-answers-to-online-security-questions-1821628011` *(last accessed 14.01.2018)*

3.3 Guiding and Aiding Users

Since secure behavior does not come natural to users, there have been many attempts to make it more accessible to them and relieve the tensions between usability and security at the same time. Although systems can be changed in many ways, it is difficult to change the user. If the user needs to be kept in the loop, e.g. because constraints dictate so, we need to support them well and make careful decisions about how to support them [62]. In the following, we highlight approaches not only to make systems more usable, but also to influence user behavior regarding passwords.

3.3.1 Password Composition Policies

Combating weak passwords has received the most attention from research and practice. The idea of enforcing certain password requriments dates back to the 1970s, i.e. the early days of cybersecurity. Morris and Thompson acknowledged password authentication is inherently flawed in terms of usability. They suggested to make users either choose longer passwords, or systematically assign passwords to them to [230]. At the time, guessing attacks were not as powerful as they are today and it was obvious to shift responsibility to users [102]. When the Electronic Authentication guideline was published by NIST, password composition policies became the de-facto standard in attempting to make users select stronger passwords. The NIST guideline suggested passwords be at least eight characters long, include at least one upper case letter, one lower case letter, one digit, and one special symbol [46]. Moreover, passwords may not be taken from a dictionary with common words and not be permutations of the username. By looking at entropy estimates, it was argued that the resulting passwords would achieve at least 30 bits of entropy. Interestingly, this was not the only specification for policies. The guideline specifically says that passwords could also be graded with some other metric and be rejected based on their estimated entropy. At the time, there was not much evidence that user-selected passwords created under the NIST-policy were in fact strong, which set of a number of research studies, and made password composition policies one of the most studied topics in password security.

Proctor et al. found in 2002 that certain "proactive password restrictions" lead to stronger passwords [249]. In two laboratory experiments, they had participants create a new password for a university account. The policies differed in the required minimum length (five and eight) and additional requirements like upper-/lowercase letters and digits. In the first experiment, where passwords only had to be five characters long, introducing the additional requirements had a greater effect on strength than in the eight-character-minimum condition. Increasing the minimum length by three characters already had a stronger effect. Interestingly, they concluded "Perhaps the most important message of this study is that restrictions on user-generated passwords may not accomplish their intended goals." Proctor et al.'s early hypothesis that policies are more or less ineffective is particularly surprising because in the fifteen years that followed, many research papers were written about such restrictions and many of them ultimately came to similar conclusions. In the following some of the most influential works are summarized.

Inglesant and Sasse report on a diary study of password policies in corporate contexts [170]. They reached the disillusioning conclusion that password policies reduce employees' productivity. Once they found a password that fulfilled the policy, participants used it as a resource to turn to when creating more accounts. Similarly, Komanduri found that users barely go beyond the minimum requirements of policies [195], but some policies yield better results than others. Weir et al. categorize policies into "explicit" and "implicit" policies [350], where explicit policies have predefined rules about the password structure like the NIST policies which is explicitly based on lowercase, uppercase,

digits, symbols (LUDS) [354]. On the other hand, implicit policies focus on strength estimation and are somewhat more volatile and intransparent to the users. For instance, if the policy uses a blacklist of words, that are disallowed, the list is usually not displayed to users up front. The blacklist only becomes visible after the first attempt is made. Moreover, they used subsets of the RockYou data set that fulfilled the NIST policy. Here, Weir et al. provided early pointers that many of those passwords can be easily cracked, too.

Thus, the logical next step was to find replacements for the NIST policy with better usability and security. First attempts to find it were of theoretical nature. Blocki et al. modeled an "optimal" policy as algorithmically solvable challenge [27], which was proposed by Shay [281]. However, while theoretically sound, empirical evidence was necessary to quantify the effects on user behavior.

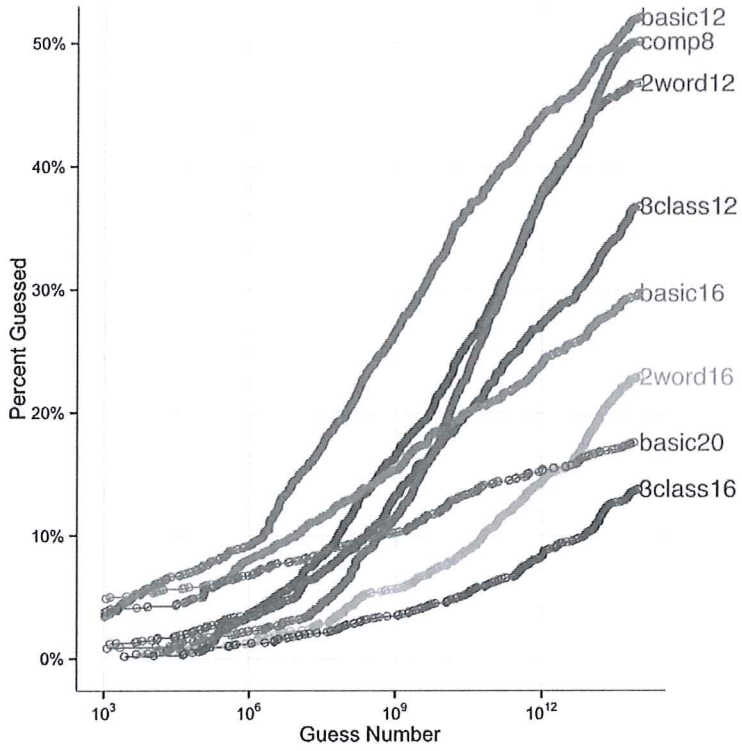

Figure 3.3: Guess number graph for passwords created under different password composition policies. A higher percentage of cracked passwords (larger y-values) indicates that passwords were weaker. In this case, passwords adhering to a basic12 policy were the weakest, while 3class16 produced the strongest passwords.

During the last few years, Shay et al. have established a taxonomy for policies [195, 286, 282]: Basic policies that only mandate a certain length (e.g. basic12), character-class centric policies that require between 2 and 4 different character classes and a given length (e.g. 3class12), policies requiring phrase-like syntax and a given length (e.g. 2word16), and complex policies that have more than 4 specific requirements (comp8). Table 3.2 illustrates the specific differences as found in the taxonomy. In multiple studies they compared the guessability of passwords under different policies. They found that a 3class16 policy produced the least guessable passwords, while basic12 yielded the highest cracking success rates. For a limited attacker, who can make up to 10^6 guesses, basic16 and basic20 performed poorly, but they fared well at the cut-off threshold of 10^{14}. The NIST-policy (comp8) performed well up to 10^6 guesses, but almost as many passwords had been cracked after 10^{14} guesses as for the basic12 policy. This refutes the postulation that character diversity automatically

leads to stronger passwords when users select them. Shay et al. point out that 28% in the comp8 condition only fulfilled the criteria by adding an exclamation mark "!" at the end, which is corroborated by Ur et al. [330]. So, as hypothesized, policies significantly influence the strength of user-generated passwords. In terms of usability, there are signficant differences, too. Shay et al. examined typical usability metrics, as well as user ratings. basic12 passwords were the easiest to create, fastest to type and easiest to remember. Interestingly, the 3class12 policy was comparable in all dimensions. All in all, basic16, 3class12, and 2word16 seem like the "winners" in terms of security and usability, but 2word16 passwords have high beta guess rates [286]. To combat this, Shay et al. suggest blacklisting specific sub-strings [282].

Table 3.2: Example policies from CMU taxonomy.

Policy	The password needs to ...	Example passwords
basic8 (1class8)	be at least 8 characters long	password monkey123 qwerasdf
3class12	be at least 12 characters long and include three different character classes (upper, lower, digits, symbol)	Password1234 2MonkeysBite NfJidl2kdils
2word16	be at least 16 characters long and include at least two letter sequences that are separated by a non-letter sequence.	password.unlocks 1-Monkey-Bites qwer.asdf.zxcvb.1234
comp8	bet at least 8 characters long, include at least one character from each character class, and not include a dictionary word	P@ssw0rd !M0nkey1 LGtjj{Rd;w1u/

In the industry and public institutions, password expiration policies are commonplace [170, 52]. Forcing users to reset their users in predefined intervals was argued to mitigate threats arising from data breaches. If password leaks go unnoticed, at least they expire at regular points. For users, an expiration policy drastically increases password overload and memory interference, for which the typical coping strategies (see Section 3.2) prevail: an expired "password1" quickly is reset to "password2" and so forth [372]. Thus, it has been argued that expiration is ineffective and causes users too much effort without graspable security benefits. Chiasson et al. set out to quantify the benefits of expiration but their mathematical model strongly indicated that the benefits are "marginal at best" [52]. If expiration is not enforced, security experts often advise users to change their password often. Zhang-Kennedy et al. reframe this rule to "change your password well" [373], i.e. users ought to change it once they suspect (or find out) a service had been compromised.

Looking at real-world policies, Wang et al. also find inconsistencies, i.e. not all web sites require the same password characteristics [346]. Users can get confused if multiple sites mandate different features for "security reasons", because they might wonder who is right. Florêncio and Herley compared policies of high-traffic websites and public institutions, mostly universities [110]. Surprisingly, the most influential companies enforce some of the loosest policies. The researchers argue that decisions in these companies are not only influenced by security officers (who evangelize password strength and expiration) but also by user experience experts (who point out the usability issues). In public institutions, security advisors outnumber human-factors experts, which is why policies represent an "overshoot" of security there, imposing considerable nuisance at marginal security benefits.

To summarize, policies have power to influence password behavior. However, this influence arises from coercion and thus risks reduced usability. Users often cope by picking the easiest possible

password that still fulfills the rules. This is why Florêncio et al. argue that we should not try to fix the user, but to fix the system (in this case policies) first [113]. Especially, service providers should be careful not to impose strict requirements and nonsensical policies[6]. However, if service providers feel the need to move to a stricter policy (and expire passwords at the time of policy change), Shay et al. at least provide evidence that users feel better protected afterwards, even though their password changes are well predictable [285]. Finally, NIST recently recognized that errors were made in the Electronic Authentication guideline and released an updated version @ref. William Burr, who was the lead author of the original policy recommendation, was recently quoted to regret his contribution. He told the Wall Street Journal "*It just drives people bananas and they don't pick good passwords no matter what you do*"[7]. Shay et al.'s results, however, somewhat relieve Burr from his remorse, because some policies help people pick "good passwords".

3.3.2 User Education and Guidelines

Policies can be considered a means to "enforce recommendations". However, giving advice to users to educate them about password security is a softer approach. A Google search for "password recommendations" yields ≈ 113,000,000 results, some very brief[8], others very elaborate and authoritative[9]. We summarized common advice and the characteristics of "bad passwords" in Section 2.4. The original NIST guideline also read more like a "how to" than a dictum for policies, although it was translated into policies after all. Sasse et al. expressed doubts about the effectiveness of user education, because it "will only work if users are motivated" [266]. We know by now that security is a secondary goal, and thus it is unlikely that people are motivated to educate themselves about it. Therefore, we have to focus on the restriction "if users are motivated" and present advice effectively in this opportune moment. Security can become the primary task, or on-par with the primary task. For instance, the moment users adopt a password manager and have to set-up their master password, they not only want to simplify password management, but also ensure that their central hub is safeguarded against attacks. Some web-accounts are of great value and users are potentially more open to receive support to secure those.

We can identify four central problems with password advice. 1) There is no consensus about adequate password strength, because it always depends on the attack model. Thus, any guideline should differentiate between threat models and brief the users about them. However, this is rarely done in practice and most advice is opinionated [162]. Besides, password advice becomes outdated if new threat models prevail. 2) Reading a guide does not necessarily translate into action. Herley says users are rational in rejecting advice if it entails too much effort [160]. Forget et al. empirically showed that advice can lead to insecure behavior, too [120]. 3) Users misread password advice. Ur et al. argue that users misconstrue a statement like "adding digits, upper-case letters, lower-case letters and symbols add to the strength of the passwords" to "*all* passwords with digits, upper and lower case letters, and symbols are strong" [327]. Heterogeneous composition policies, and feedback can influence mental models, too. Forget et al. argue that one needs to understand users' mental models of authentication first, before they can be effectively instructed [120]. 4) Lastly, if password advice achieves to change user behavior on a larger scale, password guessing attacks will be modeled around

[6] An amusing collection of nonsensical policies is collected on `https://twitter.com/PWTooStrong` *(last accessed 14.01.2018)*

[7] `http://fortune.com/2017/08/07/password-recommendation-special-characters/` *(last accessed 14.01.2018)*

[8] `https://www.ibm.com/support/knowledgecenter/SS42VS_7.2.7/com.ibm.qradar.doc/c_qradar_niap_password_recommendations.html` *(last accessed 15.01.2018)*

[9] `https://www.ncsc.gov.uk/guidance/helping-end-users-manage-their-passwords` *(last accessed 22.12.2017)*

the recommendations, too [163]. Thus, in the long run, advice needs to be revised because attacks become too efficient.

To conclude, one needs to stay realistic about what can be achieved with password advice [115]. Zhang-Kennedy et al. provide this realistic view on advice [373]. They revised the "character diversity" recommendation and suggest not using common passwords, predictable substitutions, or dictionary words. However, we still face the problem of communicating to users what "common" passwords and "predictable substitutions" are. Zhang-Kennedy et al. propose users should come up with original mnemonics, which is a special kind of advice discussed in the following.

3.3.3 Password Selection Algorithms & Memorization Techniques

Many researchers have proposed techniques and algorithms to help users create memorable and strong passwords. Haskett put forward PassAlgorithms [155], where the user remembered how to respond to a challenge rather than a static password.

Mnemonics and Training Barton and Barton were likely the first to propose mangling strategies in this context [16]. For instance, they suggested to use sentence-based mnemonics and mentally connect passwords to different cities. For Paris, a sentence like "I love Paris in the Springtime" would translate into "IlPitS". This work was seminal and some of its techniques persist in password recommendations today. Yan et al. empirically showed the benefits in terms of memorability and security of the resulting passwords [364]. However, Forget et al. observed that telling participants to generate phrase-based passwords can be misinterpreted and more guidance is necessary to achieve benefits [120]. Maqbali [215] and McEvoy [225] recommend using contextual or site-specific cues on websites as mnemonics.

Memorization by repetitive training has also been suggested. Bonneau and Schechter put forward solution that is supposed to help users memorize "56 bit secrets" (for comparison, the original NIST guideline demanded 30 bit) [35]. Users first pick a self-selected password. They system then displays a random code (or words) for each user at login-time which needs to be typed correctly into a separate field. The code becomes part of the password and was displayed with increasing delays. But participants could skip the delay by entering the code from memory. After a median of 36 log-ins 96% of participants had memorized a 56 bit secret. Despite the high success rates, it is questionable if this type of aide is in the users' interest, especially if it is deployed by more than one service. In a similar vein, Kroeze and Olivier proposed using gamification to make the training more enjoyable [199].

In terms of security, mnemonic passwords are predictably based on common phrases from movies, literature, songs, etc., but stronger than intuitively selected passwords [201]. Yang et al. demonstrate that even mnemonic phrase-based passwords can be attacked easily [365]. Thus, like Zhang-Kennedy, they highlight the importance of original, personal phrases and show how users can be instructed effectively.

Passphrases Another technique proposed to create strong, memorable passwords is to create passphrases. Rather than taking single letters from a phrase, we understand them as a combination of words, e.g. the paragon "`CorrectHorseBatteryStaple`". Passphrases are usually longer than traditional pass*words* and thus increase password strength. The PGP system uses passphrases to encrypt private keys on the clients. Keith et al. showed that, at the same time, passphrases are more memorable than more complex passwords, especially if they include punctuation symbols [186]. However, participants in their study made significantly more errors, which is the biggest usability caveat of long passphrases. On devices where text-entry is cumbersome (e.g. on mobiles or smart TVs), refraining

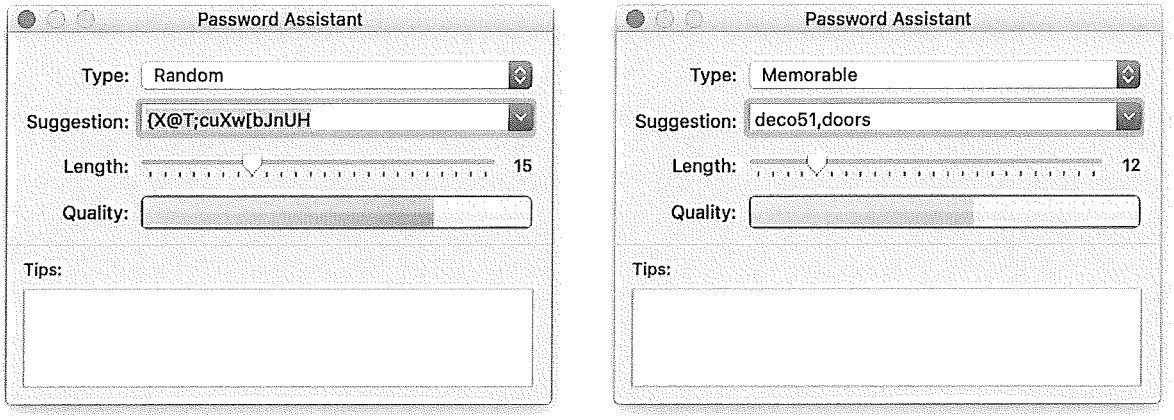

Figure 3.4: On macOS, the Keychain application manages passwords for the user. If a new entry is manually created by the user, the password assistant can be used to generate different kinds of passwords: Random (image on the left), letters and numbers, numbers only, memorable (image on the right), and FIPS-181 compliant

from masking entry can mitigate this problem [226]. Paradoxically, passphrases sometimes exceed character limits for passwords on certain websites [47].

Moreover, Bonneau showed that user-selected passphrases are predictable [36]. To mitigate predictable word combinations, system-assigned passphrases have been evaluated. Shay et al. conclude that this boosts strength, but users dislike them [284]. Similarly, policies focused on passphrases (e.g. 2word16) lead to passwords with high guess numbers, but users struggle to create them [286]. The Diceware approach is often mentioned as a means to randomize word selection [10]. The user rolls a dice firve times and notes down the resulting numbers. Afterwards, they look up the word with the corresponding number from the Diceware wordlist. The process should be repeated at least twice to create a passphrase. Unfortunately, the process requires some dedication, time, and a dice. Nonetheless, passphrases usually do not require a change to the provider's system, which makes them a viable, memorable and secure option, e.g. if they are mostly entered with a physical keyboard.

3.3.4 Password Managers and Generators

Many of the problems around password security originate from the predictability of user choices and preferences. Therefore, systems that remove the decision-making process for a large part seem to remove weak passwords. Password generators can create (pseudo-)random strong passwords for the users and thus the element of human bias vanishes. There are different approaches to generate passwords for users under two paradigms: easily memorable passwords, or passwords requiring external storage (see Section 3.2.3). In the first paradigm, the generator creates pseudo-random strings resembling user-selected passwords. **Pronounceable** password generators do not yield random characters but random phonetic segments [128]. This allows users to repeat passwords in their head or saying them aloud to memorize them. However, generating passwords that are consistently pronounceable is challenging[355, 133]. Passphrases can also be generated in the same way [202]. The second option yields truly random character strings taken from an alphabet large enough to mitigate most guessing attacks (see example in Figure 3.4). Huh et al. propose letting users replace a certain number of characters to create a more memorable version of the random password [168]. However, memorization

[10] `http://world.std.com/~reinhold/diceware.html` *(last accessed 16.01.2018)*

of random strings is still exceptionally tedious for most users, as is typing them. passwords need to be either be written down or stored inside a password manager. managers (see Section 3.2.3), offer to generate passwords. Moreover, the Safari bro randomly generated password to users and stores it into Apple's cloud storage Keych a password manager gives users a number of usability benefits [368]. As mentioned reduces interaction times and recall problems. If passwords are generated on the fly, the effort create passwords is removed. It is easy to generate unique passwords for each account and thw phishing attacks, so the PWM scales with the number of accounts. Some PWMs give the users fee back to assess their overall protection level (see Chapters 8 and 12). On the downside, users becon dependent on the PWM, which makes it difficult to roam and use other devices. Even if users do n generate passwords, they lose muscle memory with auto-fill, so it is cognitively more challenging log in manually.

HCI researchers have tried to solve these problems. Stobert and Biddle proposed VersiPass [30 It uses graphical authentication and hints to avoid losing muscle memory, but it has not been er pirically evaluated. Tapas is a decentralized password manager based on two-factor authenticatio which was well received in two user studies [223, 222]. Yee's PassPet is a browser extension that le users pick a "pet name" (label) for each website they have signed up for [368]. The label is part the password hashing and aims to increase both the security and memorability of the scheme, simil to the Password Multiplier system by Halderman et al. [145]. Fagan et al. investigated the reaso for (not) adopting a password manager [100]. They found that users more prominently apprecia usability benefits. Those who do not use a PWM distrusted the security, potentially due to a sub-p mental model. We can also hypothesize that many users want to stay independent and not give aw control to a third party.

In summary, password managers aide users with password selection, and scaling the increasir number of accounts. Generated passwords are mostly a go-to method for more proficient, securit aware users who actively seek to strengthen their passwords as much as possible. While the academ research community has not been able to create PWM solutions with widespread adoption, mar commercial solutions exist. However, the adoption rates indicate that current systems have not be fully adapted to the masses.

3.4 Persuasive Interventions

Persuading users to behave differently is the last line of attack we shall discuss in this part. Usir technology to persuade users can be traced back to Fogg's seminal work on "captology", which w later recoined under the umbrella term *persuasive technology* [117]. He defines it as "*interacti computing system designed to change people's attitudes or behaviors*". Persuasion itself is seen "an attempt to change attitudes or behaviors or both (without using coercion or deception)". Th latter part further distinguishes persuasion from manipulation, where people are not aware of t manipulator's intentions. Password composition policies exert authority and coerce users to follo the rules. The need for autonomy as part of the Self-determination theory [263] is thus undermin by policies. User education is voluntary and misses its goal because users seldom decide to educa themselves on password security. Therefore, persuasion could fill the gap of providing transpare education by making alternative behavior more salient [121]. In the context of this thesis, we th understand persuasive technology as an *enabling* technology that adequately supports users whi respecting their preferences. Persuasion in HCI has become an essential topic with numerous pape

published at top-tier conferences. Hekler already highlighted the rising interest already in 2013 [158]. What is more, persuasion is one of the central topics among the top-5 most cited CHI-papers of the past five years[11]. Consequently, we regard it as highly promising direction for password research.

3.4.1 Background

In the following, we explore how to use persuasion to create "soft paternalistic interventions that nudge users toward more beneficial choices" [1].

Terminology in Persuasion and Behavioral Economics

In the design of persuasive technology, we often encounter the concept of "nudging" people, i.e. figuratively giving them a small push to act in a certain way [317]. The "choice architect" decides on the direction of the push [318]. For instance, by setting clever *defaults*, people are relieved of making an active decision and can just accept the default. Nudging strategies as part of "soft paternalism" or "libertarian paternalism" stem from the field of behavioral economics, which studies "how individual, social, cognitive, and emotional biases influence economic decisions" [2]. In other words, behavioral economists embrace the idea that people sometimes act irrationally when making economic decisions. For example, people are significantly more likely to purchase a glass of jam, if there are only six options instead of 24 – the so-called "choice paradox" [172]. As we have seen in Section 3.2, password coping strategies ultimately involve such economic decisions: Given a risk, one needs to assess the severity, likelihood, costs of mitigating the risk, and the effectiveness of the protective measure [273]. Thus, theories from behavioral economics are a useful resource that can explain user behavior.

Referring to the dual process theory, Kahneman argues that many sub-optimal decisions originate from System 1 which is responsible for intuitive and automatic thinking processes [182]. System 2, on the other hand, is the rational and effortful part in our thinking processes. He explains that most of our thinking is carried out by System 1, because it would be impossible to put the same amount of effort into every decision that we make (e.g. it is unnecessary to weigh the pro's and con's in answering "should I brush my teeth today?"). In Password selection, both systems are involved, but depending on the context, one or the other is primarily responsible for the decision. For instance, users who have formed the habit to reuse one password for all accounts will do the same for the next account they create, thus the automatic System 1 is at play. However, if the composition policy forces the user to modify their password, the cognitive challenge rises and activates System 2. Persuasive technology aims to either facilitate decision-making (i.e. supporting System 1), or block automatic processes to help the user adopt a different behavior (i.e. activating System 2). For instance, intentionally introducing delays during password authentication [214] or in browser warnings [94] lead to users spending more time on the task and act more securely, probably because the time was enough to activate System 2.

Cognitive Illusions, Biases, and Heuristics

Irrational decision-making, according to behavioral economics, shows patterns around *cognitive illusions* (also "cognitive distortions") and *biases* [209]. Acquisiti et al. describe them as " systematic, and therefore predictable deviations from rational choice theory" [1]. The resulting behavior is neither erratic, nor irrational, but may strongly influence judgment under uncertainty [323]. To still be able to make decisions under uncertainty, people utilize *heuristics*, or mental shortcuts in decision-making.

[11] https://scholar.google.com/citations?&view_op=list_hcore&venue=6NNnGOq9_mAJ.2017 *(last accessed 17.01.2018)*

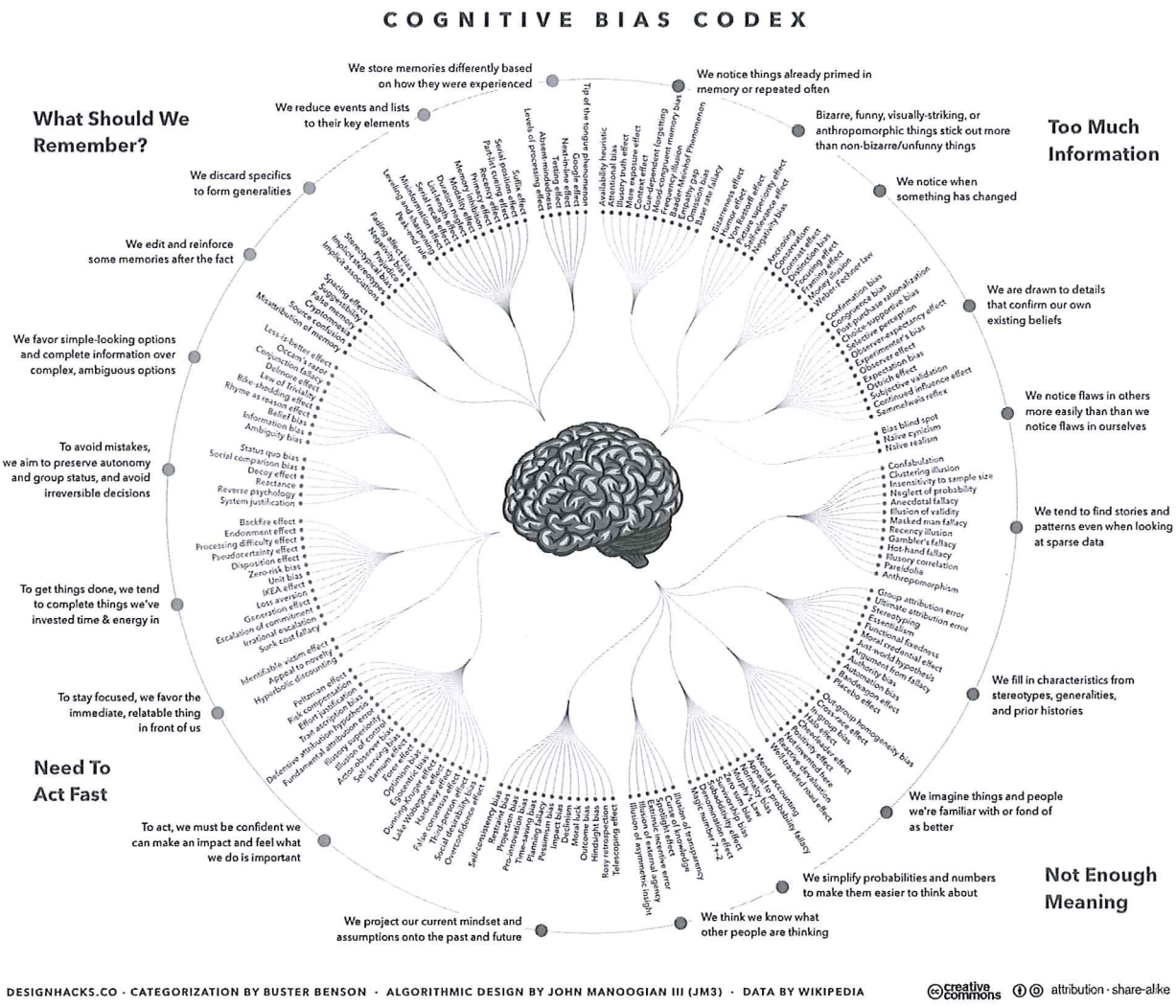

Figure 3.5: Categorization and visualization of cognitive biases ~~listed on Wikipedia~~. CC-BY-SA Buster Benson, John Manoogian III

by Buster Benson & John...

Bounded rationality explains the use of heuristics with the impracticality of assessing all possible options and outcomes. In other words, heuristics allow making good decisions under the circumstances [180]. Thus, like biases, heuristics are not necessarily bad, but even necessary to get through life [57]. The lack of certainty is prevalent in decisions in cybersecurity, especially for mainstream users. For example, it is hard for users to assess the risk of threats online, and make a decision on the required protection level. Therefore, heuristics in this domain have been investigated to better understand user behavior. The list of cognitive fallacies is long, Wikipedia mentions 109 decision-making biases alone [12] (see Figure 3.5). Let us discuss notable examples that directly relate to cybersecurity[13]. To stay on topic, we explain all effects with "password phenomena", although the scenarios have not necessarily been empirically substantiated.

The **sunk costs** fallacy describes the situation when people have made an investment under uncertainty in the past and continue to stand by it, even if circumstances have changed and it would be better to abandon the commitment [316]: Imagine someone bought a (nonrefundable) ticket to watch

[12] `https://en.wikipedia.org/wiki/List_of_cognitive_biases` *(last accessed 17.01.2018)*

[13] Please refer to Acquisiti et al. [1] for an in-depth and highly valuable discussion of biases in privacy and security

I think this is fine. But an external reviewer might be provoked by a reference to Wikipedia. Maybe this is not necessary at all?

a movie at a theater. They dislike it from the start and after 25 minutes, they feel the movie is not going to get better. Due to the sunk costs of the ticket and unrecoverable effort to go to the theater, it is likely that they stick around instead of simply leaving the show. Herley et al. hint a similar scenario for replacing password authentication [163]. In their point of view, service providers have invested into passwords and are now reluctant to move away to alternative schemes, although they might improve system security (see Section 2.5). Similarly, users who have committed to a password and reused it many times, might see the time spent as sunk cost. Thus, they do not switch to a more secure alternative even if they know there are plenty. Arguably, the **Status Quo bias** plays a role in both examples, too: people tend to favor a pre-existing state of affairs over potentially uncertain changes.

The **availability heuristic** is observed when people over- or underestimate the likelihood of events after being exposed to salient information. For instance, users might estimate the likelihood of being hacked as higher after someone in their social circle reports such an incident: the possibility has become more salient and available, thus future occurrences are seen as more probable. Das et al. partially confirmed such behavior through user interviews [68]. Certainly, **recency illusions** amplify the availability heuristic, i.e. a user who just found out about the "hacking threat" might think that this threat had just emerged. Such **salience** can be found at pass

The **anchoring** bias creates reference points (or baselines) for decisions, especially comparisons [121]. After a password breach, media reports typically mention some of the most-used passwords. Readers of news article anchor on the strength of obvious passwords like 12345 and compare their own. In many cases, their own password(s) seem much stronger by comparison, but might not be seen as strong if assessed separately. Similarly, if many people show insecure password behavior, this may reinforce an individual's practices through the **bandwagon effect**: "if so many people behave insecurely, why would I act differently?" [5].

People can also be biased by the way information is presented, which is known as **framing effect** [335]. The wording is key here and can generate **preference reversal** and other inconsistencies [165]. For password authentication, framing can be found in educating users about passwords. Zakaria et al. [371] suggest instructing users through real-world security comparisons. Framing consequences for users around losses may trigger **loss aversion** – a tendency to perceive losses as more valuable than equivalent gains [12]. Garg and Camp argue that security is currently framed as a *definite* loss for end users, while "the risk of not investing in security is a *probable*", thus uncertain, loss [127]. Cialidini shows the importance of crafting *normative* messages, that refrain from emphasizing that a socially unacceptable behavior is still widespread [56]. For instance, a normative message after a policy rejects the user's first-chosen password might explain: "The password is weak, because it is easily guessable for hackers" (emphasis on the adversary) rather than "..., because it is something many people would choose" (emphasis on the social in-group). If normative messages are framed badly, this can lead to the **backfire effect**, i.e. people behaving in the opposite way as intended. Thus, Weirich and Sasse see framing the attacker effectively as important opportunity in for persuasive password education [353, 352].

As a final example of behavioral biases, people's motivation increases as they get closer to finishing a task. Hull coined this behavior the "**goal-gradient hypothesis**" [190]. Behavioral economists have found it to induce irrational behavior: Kivetz et al. handed out two different designs of "coffee cards", that customers can fill up with each purchase [190]. Once the card is full, they can redeem it for a free coffee. One of the designs had ten stamp-fields, while the other had twelve fields, two of which were already stamped. So, in both cases, customers received a free coffee after ten purchases. Surprisingly, more cards with twelve fields were redeemed after the trial period. People (wrongly) thought to be closer to goal, and thus were more motivated to achieve it. This "goal distance model" is sometimes used by password meters, i.e. a visualization of password strength (we discuss them

in great detail in Section 3.4.3). Password strength visualization can give the user a "head start" to motivate them to fill up the entire bar and achieve a "strong password" (i.e. a full loyalty card).

3.4.2 Persuasive Design Patterns in Usable Security and Privacy

Phenomena involving biases and heuristics are often directly translated into a persuasive strategy. It is interesting that apparently the lines are blurred between the two concepts. Anders Toxboe collected a number of persuasive design patterns on his website[14], which highlights this blend of psychology and persuasion. There is, however, no such list for patterns specifically for interventions in usable security and privacy. Therefore, we start with the discussion of general frameworks and put them into the context of security and passwords.

General Frameworks

Persuasive Design (PD) has been tried to formalize in several frameworks. Fogg's Behavior Model posits that three elements must converge to achieve a target **B**ehavior: **m**otivation, **a**bility, and **t**riggers (B=mat) [116]. For passwords, this implies that people need to be motivated to protect themselves, at the same time the need to know how to achieve protection and be exposed to a trigger. Triggers work best when presented at the *opportune moment* (or) [210]. Lockton et al. proposed the "Design with Intent" framework as a toolkit for persuasive interventions [210]. Jameson et al. show how to pick strategies from various toolkits to create a persuasive choice architecture [177]. Cialdini's generic "**six weapons of influence**" have been used in the design of persuasive technology [57]. He lists authority, scarcity, liking, social proof, commitment & consistency, and reciprocity as the principles of persuasion. Let us walk through their potential usage in password authentication. The authority principle says that people are likely to follow the instructions of an authority, which explains why the NIST guidelines have been widely adopted despite their downsides. Scarcity drives motivation to act quickly to avoid losing an opportunity (see loss aversion), and framing effects are used to communicate the scarcity of a resource. For instance, the "419 scam" tactic in phishing emails frames time as a scarce resource: "your account has been intruded, if you don't log in now, you lose access to it permanently" [298]. In this case, persuasion is used for malicious purposes. People prefer to comply if they *like* their counterpart. In theory, this would make password requirements less bothersome if used on a website that users like, or in a more aesthetically pleasing way. Although there is no empirical evidence that the principle works for registration forms specifically, other HCI research points in this direction [321, 118]. The social proof strategy makes other people's behavior more *available* because in decision-making people tend to copy others' decisions. For instance, it might help to point out that millions of users are already using a password manager in a news article about a new password leak. We can also apply Fogg's behavior model here: Reading about a password leak *motivates* users to protect themselves, the social proof strategy acts as a *trigger*, and mentioning the ease-of-use of password managers gives users confidence about their ability to adopt a PWM. The commitment & consistency principle is an immediate part of the Password Life Cycle [303]. People try to be consistent with their past behavior and their past values. Thus, a small step towards breaking an insecure habit might, in fact, induce more secure behavior afterward. If a policy forced a user to stop using a common password, the effort to commit to the new password might spread over to other accounts, too. Stobert found that cybersecurity experts show higher signs of consistency [304]. Thus, if another persuasive strategy achieves that users commit to a new password selection scheme, their behavior might become more consistent and thus be elevated to the expert level. Finally, reciprocity describes the desire to return a favor. To use it as a persuasive design strategy, one has to

[14] `http://ui-patterns.com/patterns` *(last accessed 21.01.2018)*

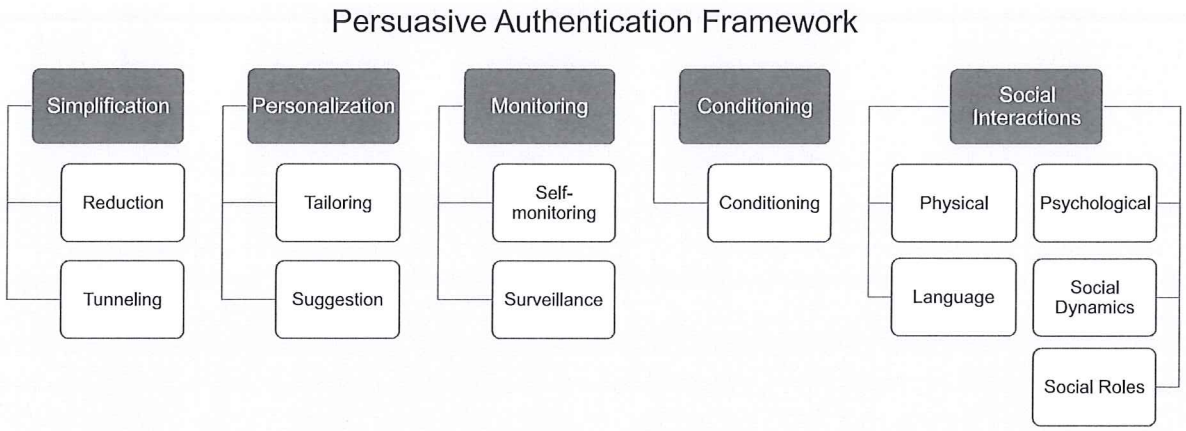

Figure 3.6: Elements of the persuasive authentication framework.

do something favorable for the users first. Thus, offering users to store their passwords in a secure place and automatically logging them in is a good foundation to ask them later to turn on 2-factor authentication to help secure their account.

The Persuasive Authentication Framework (PAF) by Forget et al.

Forget et al. embraced persuasion as technique to help users act more securely in password authentication [121]. In 2007, the put forward the Persuasive Authentication Framework (PAF), which breaks down the dimensions of persuasive design specifically for password support systems. In the following we describe it on a high level and illustrate its components with results from empirical studies.

The *simplification* principle posits to reduce the number of tasks to achieve an overall goal. Thus, the hypothetical distance between start state and goal is shorter, so, goal-gradient effects become visible. Password managers usually simplify authentication by reducing the best-case interaction from Recall-Enter-Submit to Submit. Fagan et al. showed that simplification is the primary reason to adopt a password manager [100]. But even without a PWM, both the "recall" and the "enter" subtasks can be simplified. Recall is facilitated with consistent password strategies. Entering is easier the primary password is replaced by a less complex one-time password.

Personalization, as understood in the PAF, encompasses tailoring the experience to an individual user, or to suggest actions. A recently (**@ref**) emerged design pattern spreads out the log-in process across two separate steps for username and password, rather than having the two input fields visible at once. After submitting the username, the page is *tailored* to the user by showing them their profile picture and/or another piece of non-critical personal information. At the same time, this establishes a light-weight trusted path [367]. Wilkinson et al. suggest personalizing privacy by design by giving users different levels of control to depending on their privacy profile [360]. Besides, this is a good example of the paternalism in persuasive interventions, because the designer what option is "better" for the individual user. Similarly, it is conceivable to personalize password policies depending on the user's "password profile" (see 13.1). Moreover, personality constructs have been investigated to tailor user experiences, which perhaps exceeds the original proposition in the PAF. Recently, Egelman and Peer advocated the use of psychometric cues to contextualize privacy or security messaging, which they call the "next frontier in privacy and security research" [97]. Jeske et al. investigated user profiles for the susceptibility to security nudges [179]. In their experiment, they used *salience* to steer users away from insecure wireless networks. They found that participants with low impulse control were

more susceptible to nudges. The finding can be interpreted that nudges directed at System 1 seem more effective. There is also preliminary evidence that susceptibility to phishing, i.e. a social nudge, is also related to personality [146]. It stands to reason that password selection is related to personality, too (see Chapter 7). If this were the case, a new spectrum of personalization strategies arises. Haque et al. have already proposed a psychometric construct to identify such associations [151]. Forget et al. list *suggestion* as part of the personalization dimension, because suggestions might be based on a personal context factors. They provided a number of cases studies on suggestive password alterations [123, 119, 124]. The "persuasive text passwords" (PTP) system takes a user's password and either randomly inserts characters or replaces existing characters to increase strength. The high resemblance to the original password is supposed to maintain memorability [119]. Shay et al. later re-evaluated the approach in an MTurk study, which yielded mixed results [287].

Arguably, observing what users do increases the chances that they comply to security policies, given that users are aware of being observed. *Monitoring*, however, can entail reduced user experience (cf. need for autonomy [263]). Nevertheless, if used adequately, it can serve as *authoritative* strategy (cf. Cialdini's six weapons of influence [57]). Google's password alert[15] is a moderate approach to secure users' Google accounts. It is a browser extension that alerts the user if they type in the password of their Google account on unrelated web pages. Hence, the idea is to make sure users use a unique password for their Google account, which is a central hub for various services. To work properly, the browser extension needs to access every keystroke. Some users might distrust it because, although it is open source, it is hard for them to tell if their private data is collected. Prospect theory tells us that in this judgment under uncertainty, losing private data is attributed a higher value than the security gain. A less privacy-invasive design element that might help in relieving uncertainty is *self-monitoring*. Self-monitoring through *feedback-loops* – another design pattern [206] – allows users to reflect on their past actions and derive alternatives to attain their goal. Some PWMs create a "security score" for each user and persuade them to improve it (see Table 12.1 in Chapter 12).

Users rarely develop habits to act securely. Thus, Forget et al. argue to use various forms of reinforcement to persuade people to develop such habits. Rewarding users for secure password behavior serves to *condition* them. Google Drive, for instance, offered users more cloud storage quota if they completed a two-minute security check-up [16]. Positive feedback during password selection can condition users, too. Other rewards, e.g. longer session expiration or faster system response, have not seen significant uptake.

Lastly, the PAF includes *social interactions* as persuasive design strategy. Forget et al. posit that authentication systems mimick, e.g., the users' language to convey "team spirit" [124]. Weirich and Sasse [353], respectively Sasse and Flechais [267], similarly describe authentication as *socio-technical system* that follows a social protocol. DiGioia and Dourish formulated the *social navigation* pattern [85]. To perceive the system as capable communication partner, signs of previous interactions inspire trust. Egelman et al. tried to nudge users during password selection through the *social proof* strategy [98]. A visualization informed the study participants how well their password fared compared to other users, e.g. "your password is stronger than that of 85% of our users". They did not find evidence that this persuasive strategy influenced people, but maybe the approach could have been more focused on the *proof* aspect, rather than *competition*. The normative message can be read as "other people's passwords are bad, but many people act this way". This could actually evoke backfire effects through social proof: although users see their password is stronger, they believe that the social norm is to pick weaker passwords, which makes them conform to the social norm. In fact, Weirich and Sasse have provided empirical evidence for such behavior [353]. Social interactions are perhaps one of the most powerful design elements to persuade users: Das et al. reported that radical behavior

[15] `https://github.com/google/password-alert` *(last accessed 20.01.2018)*

[16] `https://twitter.com/googledrive/status/697104410296455168` *(last accessed 20.01.2018)*

changes often occurred due to social processes [68]. They argue that it is critical for users to observe experts in their social circle to raise their awareness and motivation for cybersecurity. Thus, persuasive design could make expert behavior of known peers more visible

In a number of ways, some ideas from the Persuasive Authentication Framework were rather optimistic and from today's point of view they appear questionable (e.g. conditioning users like animals). Much evidence from the ten years that followed its publication shows that persuasion does not always work as intended, so maybe a few suggestions were a bit naive. Renaud et al. evaluated eight nudges that were supposed to make students create stronger passwords [257]. They reached the disheartening conclusion that none of them work. Nevertheless, certain aspects have caught on and are actively used, like Google's Password Alert which is a direct implementation of Forget et al.'s ideas. We can conclude that the PAF has matured over time, with certain components receiving more weight through empirical evidence, and others becoming obsolete. Thus, a revision could incorporate our newly gained understanding of persuasive authentication.

Dark Patterns

At the outset of this chapter, we discussed ethics as critical factor in password studies. Using persuasion inherently becomes ethically problematic if the intention of the influencer is concealed, either by poor design or by choice. Persuasive techniques are used to manipulate, too. Many social engineering scams on malicious websites use persuasion techniques, too. Muscanell et al. investigated the misuse of Cialdini's weapons of influence in cyber scams [231]. They found that scammers exploit all of them in social engineering attacks. Muscanell et al. also propose mitigation strategies, but do not lay them out in detail. Attacks that take advantage of "scarcity" as persuasive strategy often rely on *fear appeals*, e.g. "Your computer is infected, download this software *now* to remove all threats". Xu et al. proposed using similar fear appeals to nudge users towards anti-spyware measures [363], and Vance et al. utilized the "time to crack" as fear appeal during password selection [332]. There are a number of "dark patterns", which Nodder dismantles in his book "Evil by Design" [236]. Some of them are still benign, others highly manipulative. DarkPatterns.org summarizes them briefly, e.g. *misdirection* - where the design aims to focus the user's attention on a decoy to plant something on them[17]. However, Sasse strongly urges to resist them in the design of password persuasion because they wear off over time and often exaggerate the risks [265]. Hence, dark patterns most likely turn out counterproductive.

3.4.3 Password Meters: Persuasion at Play

The most prominent and widespread persuasive strategy directed at passwords are password strength meter (PSM). Most commonly, they proactively estimate the strength of a password at entry time and visualize it. Often, verbal feedback accompanies the visualization. They have been used on a multitude of websites, in password managers, and as standalone tools. When they are used on a website or in a PWM, there are a number of persuasive patterns at play:
Feedback and **self monitoring**: Users are enabled to make an informed decision regarding the security of the password. **Kairos**: Meters are presented at the opportune moment of protecting an account. **Goal-gradient effect**: The closer users get to a password that is deemed strong the more motivated they become to achieve the strong rating. **Simplification**: A visual strength bar is universally understandable. **Personalization**: The dynamic visualization is user-dependent. **Suggestion**: Some meters suggest alterations if they detect weak passwords. **Reinforcement**: positive feedback about the user's password can reinforce secure behavior **Authority**: If the service provider is a trustworthy entity,

[17] `https://darkpatterns.org/types-of-dark-pattern` *(last accessed 20.01.2017)*

their feedback is more likely to influence the selection **Social interaction**: Verbal feedback can speak the user's own language and open a dialog. **Loss aversion** and **scarcity**: The strength estimation is only shown during password entry – an opportunity to learn about one's own abilities which should not be missed. **Availability** and **salience**: The strength estimation makes weaknesses more salient and threats more available. **Framing**: verbal feedback and color coding allow framing the strength in a nuanced way.

Balancing all those strategies in a particular design has received much attention in HCI and security research. It is hard to trace back their origins, but proactive password checks have been part of policies since the late 1980s [354]. Bishop and Klein developed the `pwcheck` command-line tool that was used to give terminal-users feedback on their password selection [24]. A core challenge is the strength estimation proxy. Websites cannot implement a full-fledged cracking infrastructure just for the sake of password feedback. Thus, proxies and estimators are the go-to method.

Effective or not?

A number of studies have been conducted on the effectiveness of password meters. Ur et al. explored 14 different designs, respectively settings, of password meters [328]: different variations of the "bar" visualization, different stringency parameters, suggestions, and nudges. They were able to identify significant differences in passwords strength depending on the meter. Very stringent meters led to stronger passwords. Any kind of visual meter resulted in longer passwords, so text-only feedback is less persuasive. Many participants in their mTurk study added another character at the end to receive a higher rating from the meter. Qualitative feedback and usability ratings showed that the stringent meters were unpleasant for many participants, reducing their persuasive power. Egelman et al. investigated context factors in the effectiveness of meters [98]. They found that password meters seemed to have a greater positive effect on strength, if they were displayed on a site with higher perceived value. Participants in that study did not "need" the password meter, but receiving feedback on highly values websites made their importance more available. At the same time they concluded, that the design of the meter plays an insignificant role and that social nudges do not affect strength either. Moreover, both Ur et al.'s and Egelman et al.'s study showed that memorability was not affected by the meters. The meter with the highest reported effects was recently presented by Ur et al. [326]. After carefully evaluating the design space for rhetorical framing of strength feedback [89], they isolated the components of password meters and compared their effectiveness. They found that textual feedback is recommendable to create more persuasive meters. However, the policy affects the effectiveness, too. Their final design (see Figure 3.7) uses a number of persuasive elements: A visual bar to simplify communicating password strength; text feedback about the strength; explanations about the scoring with clear calls to action (tunneling strategy); and lastly, a personalized suggestion of an alternative password.

Beyond colorful bars

Visualizing strength is not the only way to design a password meter. Komanduri et al. crafted a sly prototype to tell users their passwords are predictable: their *Telepathwords* system predicts the next character the user is about to type [194]. The rationale of the system is that users do not mindlessly enter a weak password anymore, i.e. that the feedback should activate System 2 thinking processes. Furthermore, realizing that the next character can be automatically guessed might evoke fear appeal and steer people away from predictable passwords. Komanduri et al. evaluated Telepathwords with an mTurk-study using a role-playing scenario. They found that password created in the Telepathwords groups outperformed traditional password policies in regard to strength and memorability. However,

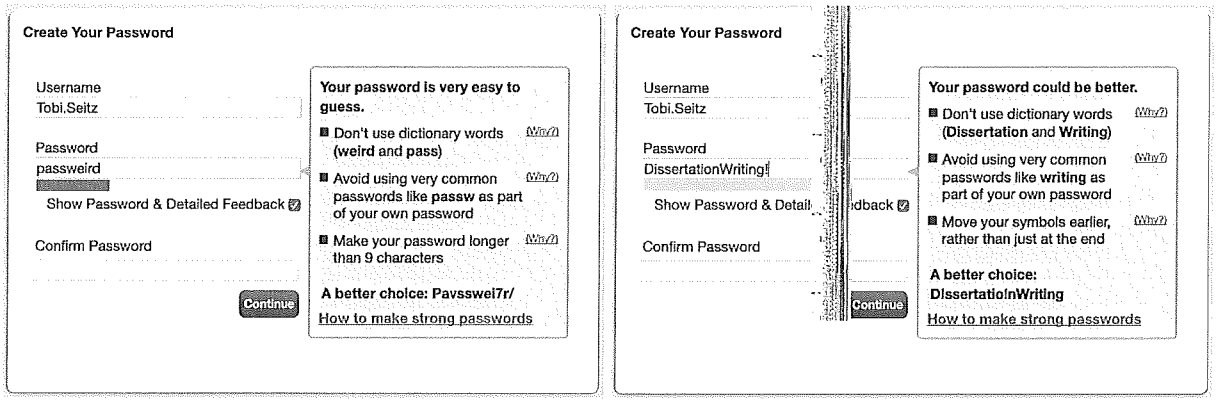

Figure 3.7

participants were significantly more annoyed by Telepathwords, perhaps because of the inconvenient truth and fear appeal. Communicating password strength with background information is another technique to persuade users. For example, Yee et al. displayed the estimated *time-to-crack* when users selected a master password for the PassPet password manager [368]. For users, it is much easier to translate this metric into a behavior than a numeric score like "3/5". Vance et al. found that such fear-appeal strategies persuade users to read password advice and modify weak passwords [332]. Khern-am-nuai et al. also measured the persuasiveness of context-based warning messages as part of the password meter [189]. They found that participants in their mTurk-study made significantly more changes to their initial choice when a warning was present, e.g. "Weak. We estimate that the password you chose is among the 30,000 weakest passwords". However, the study suffered from a few important limitations (e.g. removing all data from users who were unable to log-in after 30 to 60 minutes). Kroeze and Olivier proposed evolving a Pokemon figure as users type to visualize the growth in strength [199]. Furnell and Esmael evaluated feedback through emojis and found positive effects on the length of user-selected passwords [126]. Afjan et al. allowed users to interactively explore the visualization they had received from the password meter [10]. Ur et al. report that a dancing bunny animation that dances faster with increasing strength failed to nudge users better than less exciting meters [328]. Shay et al. tried to help users select a stronger password through a wizard that explained mnemonic phrase-based passwords [287]. Besides, they compared this strategy to the insertion approach we saw in Forget et al.'s persuasive text passwords (PTP) [123]. Both approaches were generally disliked by the participants, though.

Apart from pure strength visualization and persuasive messages, real-time feedback can accompany a password policy. The user sees a list of requirements and as they enter their password, they get feedback on the aspects that have already been fulfilled, which was originally proposed by Proctor et al. [249]. Although we know by now that the resulting passwords are not necessarily stronger, Shay et al. found that this kind of checklist can reduce user frustration with policies in general [287]. Feedback is also crucial if the policy utilizes blacklists to ban too easily guessed passwords [286]. Habib et al. evaluated Ur et al.'s data driven password meter for situations where blacklists are present [144]. They found that text feedback mitigates insecure selection especially for those users who intended to use a blacklisted password.

Light and Shadow

Evidence about the effectiveness of password meters is mixed, but Ur et al.'s strategy might be one of the most persuasive ones, because it combines many techniques that have proven successful in

different areas of persuasion [326]. So, in certain contexts, password meters can definitely nudge users towards more secure behavior.

However, where there is light, there is also shadow. Meters influence users' mental models of password strength in similar ways as policies. However, as with policies, it is near impossible to reach consensus on all parameters of pro-active measures and metrics for all contexts. De Carné de Carnavalet and Mannan showed that the estimation algorithms in real-world password meters largely differ, which is a result of the natural constraints of pro-active checks [72]. Such inconsistencies, which were also observed by Ur et al. [328], have high potential to confuse users: if their password is rated "strong" on one site, and "weak" on another, much explanation is going to be necessary to let the users find out why the ratings differ. Persuasive interventions probably fail at that moment. The authors suggest zxcvbn as the most robust password meter for websites and the KeePass meter as an alternative for offline tools. Others have addressed the issues of industry password meters. Wang et al. developed *fuzzyPSM* based on an optimized version of PCFG [345]. Melicher et al. implemented their neural network strength estimation in password meters, too [227]. Tupsamudre et al. demonstrated that a sudden surge of n-gram scores can be used to proactively detect modifications of common passwords [322]. Improvements will continue to surface, but in recent years we can at least observe a trend towards more homogeneous strength estimation due to the gained knowledge about guessing attacks.

3.4.4 Summary

Persuasive interventions help to shape user behavior and facilitate decision-making processes. Many influence strategies have been empirically shown to persuade people across a variety of domains [148]. Thus, I believe in its feasibility for the design of secure and systems. Naturally, not all interventions work in the same ways for all users. Still, persuasive technology asserts the claim to be in the "users best interest" and wants to enable them to make "better" decisions. Thus, before any intervention, we have to be sure to do the right thing, if changing user behavior and attitudes is the ultimate goal. We follow Acquisiti et al.'s definition, and find that good decisions "minimize adverse outcomes or are less likely to be regretted" [1]. With this goal in mind, we can set out to create novel persuasive strategies which are the focus of this thesis.