Designing for Fallible Humans

Jelena Mirkovic
USC Information Sciences Institute
Marina del Rey, CA, USA
sunshine@isi.edu

Simon S. Woo
Dept. of Computer Science and Engineering
Sungkyunkwan University
Suwon, South Korea
swoo@g.skku.edu

Abstract—Security and privacy solutions today are designed with an assumption of a rational user. System designers assume that the user is able to review all information shown to them, consider it along with other information they have, and user priorities, and make a conscious, rational decision in their best interest. We all know that these assumptions are wrong. Even worse, they are simply excuses for technology-centric, best-effort design. This paper argues for designing for fallible humans, taking into account human cognitive limitations, human bias and human preferences. Such design means anticipating human error and compensating for it with built-in safeguards, it means presenting information in a way palatable to humans, it means soliciting user input and working collaboratively with the user's cognitive biases and preferences. It means helping users weave security and privacy into their daily routine, and not view them as obstacles or overhead to other, more desirable tasks.

Index Terms—bounded rationality, human factor, cognitive bias, cybersecurity, privacy, passwords

I. Introduction and Motivation

Many security and privacy solutions today are designed primarily to satisfy technical needs of the systems and organizations, and pose unrealistic requirements on users. For example, users are often asked to create a unique, strong password for every system they access. Many systems implement the "strong password" requirement by requiring several character classes in the password, no repetitive or consecutive patterns, and sometimes no dictionary words. This goes directly against user cognitive needs – to remember a password user must be able to make sense of it. Random characters, a mix of upper and lower case letters and special symbols often make no sense to the user. Further, fine details of capitalization and letter replacement are too inconsequential for a user to remember them. To illustrate this, regard the string "th0se who c4n r34d Th1s R SM@RT". Since it encodes a sentence that makes sense to humans, most people will be able to remember its gist. But they will fail to recall which letters were changed into numbers and special characters, and each letter's capitalization. In fact, ignoring these details is precisely what enables us to read this string and interpret it as a meaningful message. As hard as it is to meet the strong password requirement, it is even harder to meet the "unique password" one. Users today have hundreds of online accounts, yet they are cognitively able to remember only tens of passwords [1]–[3]. Asking users to perform actions that go against their cognitive needs and preferences is a recipe for disaster, as evidenced by many studies on weak passwords [4], [5] and password reuse [6]–[8].

Passwords are but one of many examples where security and privacy design goes against users' human nature. Some others include: asking users to approve app permissions on installation, exposing fine-grained control over complex privacy settings on social networking sites, and leaving credibility assessment of online news to users. It is generally known from other realms of human existence that humans are not good at keeping track of, and rationalizing over, many minute pieces of information, and tending to boring and seemingly irrelevant tasks. Why then would we expect them to do so for privacy and security? We need a fundamentally different approach!

This paper first surveys our prior work on passwords (Section II), and highlights some design decisions we made that aligned well with human user's cognitive needs, resulting in superior performance. We then provide more examples from research of others (Section III), where aligning systems with human cognitive needs lead to improvements in security or privacy. We finish by arguing for human-centric, collaborative security and privacy design, where developers and users work together to create usable and effective solutions (Section IV).

II. Case Study From Our Research: Passwords

Many of our observations in this paper were motivated by our 5-year long work on improving password-based authentication [1], [7], [9]–[11]. Numerous researchers have worked on understanding user misconceptions about password security [12], [13] and communicating to users that their current password choices are insecure [2], [14]–[16]. Currently deployed systems usually force users to choose a strong password through the combination of strict password creation policies, blacklisting leaked passwords, and requiring frequent password changes [17]. We recognized early in our research that these security requirements may clash with the user's need for password memorability. Users choose passwords that make sense to them, and thus are easy to recall, but are also easy to guess. Our survey of literature on how human memory works, solidified our intuition that we are asking users to perform tasks that are unnatural [10]. Humans remember by association, relating new facts to existing memories

[18]. This is what makes many passwords contain names of family members, dictionary words and birthdays – these are the items that are meaningful to users. Humans further recall by reconstructing facts, sometimes imprecisely, from relevant data stored in the brain [18]. The less surprising and the less unique the memory, the harder it is to remember it precisely [19]. Phenomenon of fake memories, which people experience as real, is also widely studied in the context of unreliable eyewitness testimonies [20]. Putting these pieces together it is then no surprise that users use capitalization, numbers and special characters in a predictable manner, putting capital letters at the beginning of their password, and numbers and special characters in the end [?], [21]. It is even less surprising that users often forget their passwords [22]. We should not chastise users for these deficiencies, but instead try to design in a way that compensates for them.

A. Cognitive Limitations and Bounded Rationality in Passwords

We conducted a study with 49 USC students to understand how they reason about passwords [7]. Unlike prior work that relied on browser plugins to measure password habits [8], [23], or asked participants to design passwords specifically for the study and narrate their choices [12], [24], our study focused on mining existing account information from messages in participants' Gmail inboxes. This approach enabled us to find not only frequently used accounts, but also those that are used rarely. Our study was carefully designed in stages, to allow us to measure potential discrepancies between users' intentions and their actions as relating to passwords. We first displayed to a participant all their accounts we found, and allowed them to edit this list, and to label each account as important to them or not, and as frequently visited or not. We next asked each participant to narrate their password habits and their general strategy for their password choices. Next, we selected twelve accounts of different importance and access frequency from the participant's list and asked the participant to log on. We captured their password in this process and transformed it in a way that would allow us to detect similar passwords and analyze password structure (e.g., presence of dictionary words, presence of numbers, etc.) but would prevent us from learning the actual password. Finally, if we found that passwords were reused verbatim or with slight changes, we asked the participant to explain the choice to reuse.

As expected, we found large amounts of weak and reused passwords. Out of 621 accounts, 12% were vulnerable to online guessing attacks and 90% were vulnerable to offline guessing attacks. Further 98% of passwords were reused verbatim, and 100% were reused with slight modifications (e.g., adding a trailing number, or changing a few characters). What was surprising was that these bad password habits did not originate from users' ignorance about password attacks. Most participants were reasonably well-informed about password security and they had good intentions to minimize their risk. For example, 82% of users understood dangers of password reuse and 72% reported they intended to create strong passwords.

Yet, user intentions did not align with their actions. For example, user-narrated password composition (e.g., use of special characters) matched their actual strategy only 24% of the time. Also, 5 out of 49 participants (10%) said they shared passwords only among accounts they do not care about. We investigated this claim by examining important and non-important site passwords for these participants. In all five cases these participants shared a password between at least two important sites, and they also shared a password between an important and a non-important site. Users also had trouble estimating the number of online accounts they had. On the average their estimate was six times lower than the number of accounts we mined from their GMail inboxes.

What was the reason for users' bad password habits? Our findings point to the innate human difficulty to keep track of many different pieces of information and to rationalize a sound strategy around these. Users could not correctly estimate how many online accounts they had, on the average they estimated 15 accounts but our GMail inbox scan revealed around 80 – six times more! Users further believed their password strategy was better than it actually were, e.g., a unique password for each important site. Further, participants had clear and strong preference for memorability over security. When asked to explain their unsafe choices, 100% of 49 participants who reused a password verbatim said they did it for memorability. Further, 44% said they were knew about password-reuse attacks, but continued to reuse because memorability was more important to them than security. Similarly, 28% of our participants said they willingly create weak passwords, because of their memorability and convenience. We also investigated if password managers helped users create strong or more diverse passwords, but found that this was not the case. We suspect this is because users that use password manager still want to remember their passwords, perhaps anticipating that the manager may not always be available.

In retrospect, we should have anticipated the results we obtained. There is plenty of literature in public health and finance that highlights similar behavior patterns. For example, people tend to overestimate how hard they exercise [27] and how much they save when switching to energy-conservation plans [28]. They also underestimate how much they will need in retirement [29], and how much they spend on subscription services [30]. What do all these activities have in common? They ask people to enact daily changes in their life habits or make daily decisions that add up to a bigger goal. It turns out that people are notoriously bad in keeping track of how their actions add up to that desired goal. They overestimate the impact of their actions and underestimate the obstacles on their

Row	Measure		LEP	Passwords	Security Questions
1	Participants		44	93	-
2	Passwords		440	930	-
3	Statistical strength (avg)		10^{24}	10^{11}	-
4	Recall (1 week)	all-fact	31.6%	26%	32.1%–83.9% [25]
5		five-fact	47.7%		
6		four-fact	70.0%		
7		three-fact	82.1%		
8	Long-term Recall (3-6 mo.)	all-fact	16.5%	9%	6.4%–79.2% [25]
9		five-fact	33.9%		
10		four-fact	53.0%		
11		three-fact	66.5%		
12	Reuse	identical	3.1%	5.7%	
13		similar	15.4%	31.6%	
14	Acquaintance-guessing	all-fact	0.7%	9%	17%–25% [26]
15		five-fact	0.7%		
16		four-fact	0.7%		
17		three-fact	1.3%		

TABLE I

Security, recall and reuse of LEPs vs regular passwords and security questions. LEPs performance is much better thanks to its alignment with user cognitive strengths.

Passphrase approach	Participants	One week recall	Statistical attack security
User-chosen	44	45.0%	10^{15}
User-chosen with hint	56	73.2%	10^{9}
Mnemonic-guided	51	69.3%	10^{16}
System-chosen	58	14.3%	10^{20}
System-chosen with hint	56	19.6%	10^{17}

TABLE II

Security and recall of user-chosen, mnemonics-guided and system-chosen passphrases. Mnemonics-guided passphrases have comparable security to system-chosen with hint, but their recall is comparable to user-chosen passphrases.

Password approach	Participants	One week recall	Statistical attack security
3class8	207	64.08%	10^{14}
zxcvbn	207	67.28%	10^{15}
NewNIST	180	70.78%	10^{13}
Data-Driven-Pass	203	71.43%	10^{17}
GuidedPass	219	81.08%	10^{18}

TABLE III

Security and recall of passwords chosen by users, with some strength feedback (3class8, zxcvbn and NewNIST) and with specific suggestions for improvement (Data-Driven-Pass and GuidedPass) approaches. Approaches that offer specific suggestions result in better strength and better recall, because they allow users to keep their passwords memorable.

way. They are overly optimistic that they will reach the desired goal. This fits exactly the behavior we observed with passwords.

Sub-optimal decision-making in humans relates to a broader concept known as "bounded rationality" [31]. While it is commonly assumed that humans make decisions rationally, based on all available information, to achieve some desired goal (e.g., security or privacy, saving a given amount of money, staying healthy, etc.), bounded rationality theory amends this model to introduce limitations to human cognition or decision making. For example, one can assume that humans make decisions based on a limited amount of information, or based on a subset of information selected based on an existing bias. Another way to bound rationality is to impose some limits on the decision cost, e.g., preferring a sub-optimal but fast decision, to the optimal but costly one.

B. Designing for Bounded Rationality

How do we design for bounded rationality? There are multiple mechanisms we can develop that can aid users in areas where they lack strength or competency, and allow them to make better choices. Our research has explored several such mechanisms for the password problem.

One approach we took was to design a new way to create passwords – one that built on existing memories, instead of asking users to memorize new information. In our Life-Experience Password [10] work we introduced passwords as series of answers to questions about a specific, memorable, life experience, such as a trip, a wedding, a graduation, an accident, a person or a location. A user was first asked to choose a type of experience, from their past, that they wanted to use for a password. The user then assigned a title to that memory, which would help them recall it during authentication. We then asked the user a series of specific questions tailored to

their chosen experience. The questions focused on specific facts that were consistently recalled by humans. We also selected those types of facts that had high statistical strength, meaning that there were many possible answers to the given type of question, and thus it was difficult for a statistical attacker to guess the correct answer. Selected types of facts were: names of people, names of locations, dates, objects and object features (e.g., color) and activities. Answers to the questions became the user's password. During authentication the user was prompted with the title they chose for the experience and the questions, and was expected to produce the answers to these questions.

We conducted several human user studies to evaluate strength, recall, reuse and usability of life-experience passwords, and compare them to existing passwords and to security questions [10], [11]. We also wanted to measure how easy it would be for close or casual friends to guess a user's password. In one of the studies, we asked participants to create ten passwords, for ten different servers. A participant was assigned either to the control group – creating regular 3class8 passwords [1] – or to the intervention group – creating life-experience passwords. It is known that people are cognitive misers [18], [32] and seek to minimize mental effort, where possible. We anticipated that this may lead to high password reuse. To motivate participants to choose diverse life-experience passwords we offered them topic choices at random, one at a time. The participant may accept the offered topic or advance to the next choice.

Table I, originally published in [11] shows the strength, recall and reuse of life-experience passwords, compared to the same features of regular passwords. LEPs performance far surpassed that of regular passwords and even security questions, thanks to their alignment with human cognitive strengths. LEPs had 10^{11} times higher strength against statistical attacks, mostly because they used high-strength facts and explicitly asked users to provide these facts. LEPs were reused half as often as regular passwords, thanks to our random offering of LEP topics, which then motivated users to create different LEPs for different accounts. Another way that LEPs compensated for human cognitive weaknesses was to allow imperfect fact matching. Each fact was stripped of capitalization and punctuation, verbs were stemmed and nouns were converted to their singular form. Asking users to recall all facts during authentication resulted in 31.6% short-term recall (one week) and 16.5% long-term recall 3–6 months. While this was still better than passwords (26% short-term, 9% long-term recall) we explored if we could allow users to authenticate with imperfect recall, i.e. allowing for fewer matches. This dramatically improved recall: allowing five matches raised short-term recall to 47.7% and long-term to 33.9%, allowing four matches raised these measures to 70% and 53%, respectively, and allowing three matches raised these measures to 82.1% and 66.5%, respectively. Out of these options, four-fact authentication also offered good security against acquaintance guessing (see Table I). Yet another way that LEP design compensated for human cognitive deficiencies was to ask very specific questions during password creation and authentication. In our early user studies we had discovered that humans correctly recalled gist of their answers, but had trouble specifying this gist in a consistent manner. For example, they may recall that the location of some party was Sam's house, but they may use terms "Sam's house", "Sam's place" and "Sam's apartment" interchangeably. Or they may recall that they went on a memorable trip to Hawaii with Jonathan Smith, but may refer to this person as "Jonathan", "Jon", "Mr. Smith", etc. We realized that we needed a way to consistently elicit one specific variant of the user's response. We did so by creating very specific prompts, such as asking "What was the first and the last name of the person that went with you?" instead of asking "Who went with you?".

We explored another approach to improve password diversity, while preserving recall, with our work on Mnemonic Passphrases [1]. That work specifically aimed to address password reuse. We hypothesized that users reuse passwords because it is cognitively hard to remember multiple unrelated passwords. Even if a user can memorize multiple passwords, it is hard for them to associate each password with the correct account. This information is simply too mundane, too uninteresting to the user to create a lasting memory.

Similar to LEPs, mnemonic passphrases only consisted of human-meaningful words, and we normalized these words to remove capitalization, punctuation, the tense of verbs and the number of nouns. To simulate password diversity, we displayed a sequence of random letters (5–7) to a user during passphrase creation. We called this string a mnemonic. The user was asked to create a passphrase, such that each word starts with a letter from the mnemonic. If additional words were added to the passphrase by the user, perhaps to make it more memorable, we allowed this, but stored only those words that corresponded to the letters of the mnemonic. During authentication, we displayed the mnemonic to the user again, to remind them which passphrase went with the given account. A similar approach was taken by Camp et al. in [33] to create visual, random cues to help users contextualize a new password.

In a human-user study with almost 400 participants (around 50 per a type of passphrase we investigated) we measured security and recall of user-chosen passphrases, versus either mnemonic-guided passphrases or random, system-chosen passphrases. Results of that study are given in [1], and key points are summarized in Table II. User-chosen passphrases had good recall (45% recalled after

[1] A 3class8 password must have at least 8 characters, and cover at least 3 out of 4 possible character classes.

one week) and security (compromise required around 10^{15} guesses), but could lead to passphrase reuse when a user had many accounts. System-chosen passphrases would be unique to each account, and were strong (10^{20} guesses), but their recall was very low (14.3% recalled after one week). When we displayed authentication hints – mnemonics – recall increased in both categories, but the increase was much larger for user-chosen passphrases (73.2% for user-chosen versus 19.6% for system-chosen). These results confirmed our intuition. Participants were more likely to remember passphrases they chose, as these carried meaning for the participant. Hints were also more likely to help recall, if they related to the user-chosen passphrase, because they helped restore the hidden meaning that the passphrase had to the user. Hints, however, lowered the passphrase strength by about 3–6 orders of magnitude. Mnemonic-guided passphrases, where the user was prompted to create a passphrase that matched the system-chosen mnemonic, had the best of both worlds. They had comparable recall to user-chosen passphrases (hints were always displayed for mnemonic-guided passphrases) – 69.3% were recalled after one week. Their security was close to that of system-chosen passphrases with hints – 10^{16} guesses.

While LEPs and mnemonic passphrases aimed to design better authentication approaches, in our GuidedPass work [9] we wanted to evolve user-chosen passwords towards stronger variants. Again, we wanted to work alongside users and help them mold passwords they chose into stronger ones, without sacrificing recall. Our system initially allowed the user to input any password. The password was then analyzed against statistical attacks. If the password's strength was below the desired target, the system generated several suggestions that would evolve this password into a stronger one. The user was free to choose some, all or none of the suggestions. Some suggestions asked the user to make the password longer, add another character class at an unexpected position in the password (e.g., the middle), or choose less common words. We evaluated GuidedPass in a user study with almost 1,500 participants (around 200 for each password approach we studied). We compared GuidedPass against a simple 3class8 approach (requiring at least 8 characters, with at least 3 character classes), two meter-guided approaches, which only measured the current password strength but offered no guidance on improvement (zxcvbn [34] and NewNIST [17]), and against another guidance-based approach Data-Driven-Pass by Ur et al. [16]. These results were presented in [9] and we summarize them in Table III. Guidance-based approaches – Data-Driven-Pass and GuidedPass had higher recall than other approaches and they also had better security. GuidedPass was the best, with 81% recall after one week. We attributed this success to our alignment with the cognitive needs of participants. We offered an array of specific alternatives that would evolve their password into a stronger one, and let each participant decide on modifications that they were comfortable with. This helped preserve high recall.

In all our studies users were more satisfied with our improved versions of passwords than with traditional ones. In our LEP study, 93.7% of users said they would use LEPs for high-security accounts. Mnemonic passwords were rated as highly as user-chosen passwords with regard to their ease of use, and on a 1–10 Likert scale users rated helpfulness of hints for memorability as 7.76. In our GuidedPass work, guidance-based approaches, with and without strength enforcement, were rated around one point higher than meters with strength enforcement (on 1–10 Likert scale), because specific guidance helped users choose a strong password.

C. Summary and Recommendations

In this section, we summarize our recommendations for designing authentication systems that work along with human cognitive limitations.

- Do away with capitalization, punctuation, numbers and special characters. As humans put little importance into this information, including it in passwords is counterproductive, and bound to lower recall and increase password resets. Loss in strength from losing character classes can be recouped by requiring longer passwords (e.g., passphrases).
- Do away with dictionary checks, but keep leaked-password checks. Since humans remember well those passwords that carry meaning to them, dictionary checks go against human nature. Instead, require longer passwords to achieve desired strength against statistical attacks. Leaked-password checks should be kept, since using these passwords incurs a high risk of account compromise.
- Offer memory aids to help users recall which password goes with which server. These could be mnemonics, graphical cues, revealing the first few letters of the password, reminding users of password policy, etc. Require longer passwords to achieve desired strength against attacks, given memory aids.
- Use passwords sparingly. Many online sites already implement this principle, using long-term cookies and prompting users for passwords only when they sign in from a new device. Financial sites cannot use long-term cookies, but they use SMS-based, email-based and fingerprint-based authentication in addition to, or in lieu of passwords. Users can still fall back to password-based authentication if they do not have access to their phone, email or if their device does not have a fingerprint reader.
- Help users keep track of their actions and how they align with their intent. Password managers today simply store passwords, but they could do much more. They could analyze stored passwords, and alert users when they use weak passwords or when they reuse

them. In combination with browser visit history, password managers could detect stale accounts and alert users to close them. They could also let users input a preferred password policy (e.g., share passwords among unimportant sites) and alert them when their password behavior does not align with that policy.

- Automate actions whenever possible. Boring, repetitive tasks should be automated, whenever possible. If a user is already using a password manager, the manager could always suggest a strong, random password for each new account. In fact, newer versions of mainstream browsers are begining to do so today. If a user has input a preferred password policy (see the previous recommendation), the manager could suggest the password that aligns with that policy. Users also have many existing accounts and passwords that run the risk of compromise. We should develop tools that can automatically convert these, to align them with the user's preferred password policy. Online sites should also offer an option to close one's account – very few do so today! If such option were widely available, we could develop tools that automatically close accounts that have not been accessed within some user-chosen period.

III. Case Studies from Others' Research

In the rest of this paper we provide some other examples where user actions required for security and privacy tasks do not align well with user cognitive needs. These examples relate to areas where we have not performed active research, and are compiled from published works of other researchers. We use them to highlight the breadth and prevalence of cases where developers ask users to perform unnatural tasks, which results in privacy leaks, security problems and user disenfranchisement.

A. Application Permissions and Terms of Service

One area where users are asked to process more information than they are capable of or interested in is during mobile app installation.When a user installs a new mobile app, they are asked to give permissions to the application for accessing various private data. This interaction is ineffective, for numerous reasons. First, the user is at the moment interested to install and use the application. They are not likely to fully process nor to seriously consider what they are being asked to do. Second, the user lacks information needed to make an informed choice, because they do not understand the context in which the requested resources may be used by the application, nor what functionality will be impacted if the access is not granted. For example, imagine that a user wants to install a PDF reader so they could read a PDF document. During installation they are asked to grant permissions for camera or location access. The user would like to know if they can read the document without granting these permissions. They would also benefit from knowing if there is another PDF reader application that does not require camera or location access. Asking users to do their own research to answer these questions is bound to fail, and it is a simple but ineffective way to pass the burden of privacy protection onto users.

Similarly, when users install new software on any device, they are asked to agree to terms of service. Service agreements have even worse user interaction dynamics than mobile app installation. First, they offer way more text to the user, which very few read or understand [35]. Second, they do not offer any alternative. The user either agrees to the terms of service, or cannot use the product.

There is already notable research on improving mobile app installation process. Lin et al. [36] show that there is a small set of privacy profiles that a user could choose from to ensure their privacy preferences are met, and to avoid reviewing each app's permissions separately. Liu et al. [37] build a personalized assistant that can learn a user's preferences and suggest permissions settings automatically. The same authors also explore using crowdsourcing to recommend to the user permissions settings chosen by similar users [38]. Almuhimedi et al. [39] show that offering periodic summaries of how often private data is being shared, and with which apps, can motivate users to change their permissions settings. All this research is well aligned with user cognitive needs and is aiding the users in proper ways, as evidenced by high rates of user adoption of suggestions generated by research tools (95% of participants reassess their permissions, and 58% restrict some of their permissions). Building similar approaches for service agreements, and ensuring that commercial solutions implement those cognitive aids, would result in much improved user privacy and lower burden to users.

B. Privacy Settings on Social Networking Sites

Social networking sites expose complex privacy settings to users, yet many users do not understand them nor are able to align them with own expectations of privacy [40]. Even worse, social networking sites lump all the user's "friends" together, while human relationships are very diverse, with varying levels of closeness between friends, and varying readiness to share different pieces of information. This situation leads either to oversharing [41] or to excessive guarding of information and conformity [42]. Ultimately, neither the model of friendship nor the privacy settings on social networks are well aligned with user cognitive needs. Users want to share certain information with certain groups of friends [43], and have many circles of friends, of different relationship strength and type [42]. Ideally, the user would decide per each post with which friends it should be shared. This is however unrealistic, because there is too much information for the user to process, the information is boring and the processing would be repetitive. Instead, we need tools which can learn automatically the quality of information being shared (e.g., is it private, is it positive or negative for the

user, what topic does it discuss and in which light) and user's individual preferences for sharing it. Learning users' implicit assumptions and expectations of who can and should see each post is the main challenge that should be met by automated tools. We further need social networks to expose fine-grained control over post sharing, so posts could be automatically shared only with chosen friends.

C. Fake News

A topic that has attracted a lot of public attention recently is fake news or false advertising on online media [44]. The current approach by the media is to publish content liberally and let users decide what to trust. While this gives users a lot of power, it is not well-aligned with their cognitive abilities. It is unrealistic to expect each user to critically regard everything they hear or read online, and to investigate the truth of each claim. Almost all users lack time or skills for this job. Yet the danger of propagating misinformation is great, and the consequences could be grave. The impact is possible not just in political realm, but also for public health and morale. Imagine if users in a wildfire region were served content that motivates them to ignore evacuation orders. Or if young users were steered towards suicide through videos and posts. Or if users of certain race or nationality were steered towards violence against members of a different race. On another hand it is impossible to ask media outlets to assign a "truth score" to each content they serve, as the content is vast, varied and produced by many sources.

One thing is certain, however. While we do not know what the solution is, we need automated tools to address this problem, which do not tax users to be their own fact-checkers. More research is needed to understand what kind of objective measures could be extracted about each piece of content, and how these can be used to align content with user preferences. Automated ways to compare a given content to other published content can be developed, thus saving users from doing this exhaustive research themselves. In some cases claims from authoritative organizations and recognized experts could be cross-checked with content's claims and discrepancies could be highlighted for the user. For example, content making medical claims could be cross-checked against World Health Organization's publications or against medical journals. Approaches similar to crowd-sourcing and personalizing app permissions could potentially be used here to allow users to select content that either aligns with or challenges their current views.

IV. Recommendations and Conclusions

As we highlighted above, many security and privacy solutions today place an undue burden on users. This burden does not align well with human cognitive abilities and strengths. Designing systems this way clearly does not benefit the user that is left confused, frustrated and exposed to various risks. Such system design also does not benefit developers, and product- and service-providers. While giving users a myriad of knobs to turn is an easy way out, in the long run such settings result in problems for providers and developers [45]–[47]. We end this paper by summarizing recommendations for user-aligned design of privacy and security solutions:

- Do not ask users to keep track of many pieces of information. Humans do not do this well. Instead provide ways to summarize and organize this information into higher-level, actionable takeaways. Offer these summaries to users in a user-friendly way and suggest changes.
- Do not ask users to make many small decisions about their security and privacy. These tasks are boring and users may be ill-informed about pros and cons of different choices. Instead allow the system to learn user preferences either through a training phase, or by crowd-sourcing. Allow the user to make high-level decisions (e.g., "I want to do what others like me do" or "I want to share private information only if it is critical for applications to perform their core tasks"), then have the system take actions to meet user goals.
- Do not force users to perform unnatural tasks. This is bound to result in poor compliance, poor user satisfaction, missed privacy and security goals, or all three. Instead learn what users naturally do and align your demands with user strengths.

While developers and system designers must originally make assumptions about their users' abilities and preferences, these assumptions should be continuously amended, based on user monitoring and feedback. A few areas of security and privacy offer good examples of alignment between user and developer/provider needs. Operating system and application updates used to be a manual process, but are nowadays fully automated, and this is usually the default option during software installation. User feedback is solicited only when software update is likely to be disruptive, e.g., when the device must be restarted. Overall, this results in more an more timely updates, although the process is still far from being perfect [48], [49]. Similarly, many mobile devices (phones, tables and laptops) today use fingerprint readers for user sign-in, and thus avoid cognitive burden of passwords. Some mobile applications also leverage device fingerprint readers for easier and more secure user authentication, although some security problems still remain [50].

Users today view security and privacy as a burden, a necessary evil, due to their misaligned demands on user's cognitive power. If instead developers and users collaborated to achieve the design that is both user-friendly and effective, users may view security and privacy as an enhancement of their online lives. We hope future research and development will take us in this direction!

References

[1] S. S. Woo and J. Mirkovic, "Improving Recall and Security of Passphrases through Use of Mnemonics," in Proceedings of the 10th International Conference on Passwords (Passwords), 2016.

[2] R. Shay, P. G. Kelley, S. Komanduri, M. L. Mazurek, B. Ur, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, "Correct Horse Battery Staple: Exploring the Usability of System-Assigned Passphrases," in Proceedings of the eighth symposium on usable privacy and security. ACM, 2012, p. 7.

[3] C. Kuo, S. Romanosky, and L. F. Cranor, "Human Selection of Mnemonic Phrase-Based Passwords," in Proceedings of the second symposium on Usable privacy and security. ACM, 2006, pp. 67–78.

[4] X. de Carné de Carnavalet and M. Mannan, "From Very Weak to Very Strong: Analyzing Password-Strength Meters," in Network and Distributed System Security Symposium (NDSS 2014). Internet Society, 2014.

[5] J. A. Cazier and B. D. Medlin, "Password Security: An Empirical Investigation into E-Commerce Passwords and Their Crack Times," Information Systems Security, vol. 15, no. 6, pp. 45–55, 2006.

[6] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The Tangled Web of Password Reuse," in NDSS, vol. 14. San Diego, USA: Internet Society, 2014, pp. 23–26.

[7] A. Hanamsagar, S. S. Woo, C. Kanich, and J. Mirkovic, "Leveraging Semantic Transformation to Investigate Password Habits and Their Causes," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2018, pp. 2379–2388.

[8] D. Florencio and C. Herley, "A Large-Scale Study of Web Password Habits," in Proceedings of the WWW. Banff, Alberta, Canada: ACM, 2007, pp. 657–666.

[9] S. S. Woo and J. Mirkovic, "GuidedPass: Helping Users to Create Strong and Memorable Passwords," in International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, 2018, pp. 250–270.

[10] S. Woo, E. Kaiser, R. Artstein, and J. Mirkovic, "Life-Experience Passwords (LEPs)," in Proceedings of the 32nd Annual Conference on Computer Security Applications. ACM, 2016, pp. 113–126.

[11] S. S. Woo, R. Artstein, E. Kaiser, X. Le, and J. Mirkovic, "Using Episodic Memory for User Authentication," ACM Transactions on Privacy and Security (TOPS), vol. 22, no. 2, p. 11, 2019.

[12] B. Ur, F. Noma, J. Bees, S. M. Segreti, R. Shay, L. Bauer, N. Christin, and L. F. Cranor, "I Added '!' at the End to Make It Secure: Observing Password Creation in the Lab," in Eleventh Symposium On Usable Privacy and Security (SOUPS 2015), 2015, pp. 123–140.