Computer Science and Information Systems 16(1):313–332 https://doi.org/10.2298/CSIS180328016B

Rejecting the Death of Passwords: Advice for the Future

Leon Bošnjak[1] and Boštjan Brumen[1]

[1] Faculty of Electrical Engineering and Computer Science, University of Maribor,
Smetanova ulica 17, 2000 Maribor, Slovenia
{leon.bosnjak, bostjan.brumen}@um.si

Abstract. Passwords have been a recurring subject of research ever since Morris and Thompson first pointed out their disadvantages in 1979. Several decades later, textual passwords remain to be the most used authentication method, despite the growing number of security breaches. In this article, we highlight technological advances that have the potential to ease brute-force attacks on longer passwords. We point out users' persistently bad password creation and management practices, arguing that the users will be unable to keep up with the increasingly demanding security requirements in the future. We examine a set of real, user-generated passwords, and compare them to the passwords collected by Morris and Thompson. The results show that today's passwords remain as weak as they were nearly four decades ago. We provide insight on how the current password security could be improved by giving recommendations to users, administrators, and researchers. We dispute the reiterated claim that passwords should be replaced, by exposing the alternatives' weaknesses. Finally, we argue passwords will remain widespread until two conditions are met: First, a Pareto-improving authentication method is discovered, and second, the users are motivated to replace textual passwords.

Keywords: authentication, password security, comparison.

1. Introduction

Technology has seen tremendous growth and advancement since Morris and Thompson highlighted password security vulnerabilities back in 1979 [1]. Since the mid-1970s, global computer sales have increased from 50,000 to a record of 355 million units per year in 2011. With a rapid growth in the smartphone market in the last decade, cellular phone sales quickly overtook computer sales, reaching over 1.4 billion smartphones sold in 2015 alone. This exponential advancement has been influenced greatly by the Internet, the user base of which has grown to nearly half of the world's entire population. It became the predisposition for the data explosion, with global Internet traffic exceeding one zettabyte of data per year by the end of 2016. It is difficult to estimate the amount of sensitive data stored and processed online, but frequent data breaches and the emerging challenges we face to protect this data remind us of the growing importance of security.

To this day, textual passwords remain one of the most common authentication methods. Their continuous dominance can be attributed to their diverse advantages, such

as low cost, simplicity of implementation, and convenience. In computing, they have been used since the earliest days, with MIT's time-sharing system CTSS being a prime example from the early 1960s. Several possible threats have been identified in the following decades, often through accidentally discovered lapses in system security, such as poorly implemented access control lists or vulnerabilities in password encryption algorithms. While some of these problems were dealt with throughout the years, others persisted (many systems keep their passwords in a plaintext format), or even escalated (constantly increasing computer processing power is decreasing the amount of time necessary to perform a successful key search).

Ultimately, the level of security provided depends largely on the underlying security scheme (or lack thereof) implemented by the system in question. Unfortunately, systems often do not enforce sufficient levels of security. In the past, these decisions were based on the knowledge and awareness of security officers and administrators, allowing room for human error. To mitigate emerging threats, security experts have been proposing stricter password policies gradually over the years. The extent to which these recommendations are implemented, however, still depends on system designers: A conscious attempt to shift the balance between security and memorability is often made in favor of user convenience.

Thus, a considerable part in the endeavor to protect their own personal data still lays on the shoulders of the users. Morris and Thompson were among the first authors to expose users' bad password creation habits. In response, experts in the growing field of Password Security began to raise awareness among the general population. Despite being educated about password security risks and requirements, however, users remained slow and reluctant to adapt these good practices. In the end, nothing prevents the user from selecting a password that barely meets the minimal requirements enforced by the system.

Parallel to textual passwords' multi-dimensional problems, countless novel authentication methods are being introduced in research papers. They approach the password dilemma with an argument that the textual passwords have reached the end of their useful life, and must be replaced. They expose the passwords' inability to withstand offline cracking, and the users' tendency to create and manage bad passwords, while highlighting their method's advantages. As a result, stakeholders are advocating for alternatives, despite limited empirical evidence to support their superiority in real-world applications.

The problem with this approach is that all of the above points have already been well-documented, and at least some version of the argument has been advanced over and over for the last few decades. Nonetheless, textual passwords have withstood all of these claims and do not appear in danger of being replaced any time soon. The challenges thus lie in ensuring a sufficient level of security to mitigate privacy-related incidents, while continuing the search for Pareto optimal authentication solutions.

2. Related Work

User-chosen passwords have been evaluated empirically many times over the course of four decades. In their seminal paper, Morris and Thompson were the first to expose bad

password creation habits [1]. Their work was important, not only for providing insight into the extent of the password security problem as it appeared in 1979, but primarily for asserting that the situation will get worse if no improvements are made in the future.

The next twenty years saw very few empirical studies of user-generated passwords, until the steady growth of data breaches over the years prompted Zviran and Haga to conduct a study on password characteristics and usage patterns [2]. They found out that passwords had not improved at all (almost 50% of users had passwords composed of 5 characters or less, and an alarming 80% used only alphabetic characters), proclaiming the lack of user education and decreased concern with information security as the primary reasons. They provided several directions for future research that were explored in the following years.

In 2007, Florencio and Herley conducted the first large-scale password study, analyzing over 500,000 user-generated passwords [3]. They found that, albeit the minimum password length had increased to 6 characters on average, the overwhelming majority of users still created passwords that contained only lowercase characters. Three years later, Dell'Amico et al. analyzed three sets of real-world passwords from the standpoint of several password guessing techniques, including dictionary attacks, mangling rules, probabilistic context-free grammars, and Markov models [4]. Aside from finding out that no strategy prevails over the others, they once again pointed out the users' weak passwords, stating that password policies are not guaranteed to prevent the users from making them. In a 2012 study, Bonneau developed partial guessing metrics, and analyzed a corpus of 70 million passwords [5]. They showed that, against an optimal attacker performing an unrestricted brute force attack, textual passwords provide security roughly equivalent to only 20-bit random strings. Additionally, different population appears to have a minimal effect on the distribution of passwords. On the contrary, a 2016 study by Shen et al. suggests the average password length had increased to 9.46, and that the users tended to select random characters more often [6]. They also reported that the users preferred alphanumeric over alphabetic characters in their passwords, and were more likely to select combo-meaningful data rather than single-meaningful data as their passwords. In one of the most recent big-scale studies of over 145 million passwords, however, Ji et al. reported shorter passwords (composed of less than 8 characters on average) following predictable patterns (e.g. L, D, LD, and LDL) [7]. They managed to crack up to 65% of passwords in some of the datasets, and showed that several password meters classify passwords as stronger than they really are against modern password cracking algorithms.

These inconsistent findings on the (lack of) improvement of password security over the last decades have been the cause of many disagreements in regard to whether textual passwords should be replaced or not. This question is discussed extensively by Bonneau et al. [8], who offer a multi-dimensional perspective on the password-related dilemma, including the problems with password research as a whole. One of our main contributions is expanding on the work of Bonneau et al. by providing a historical overview from the standpoint of technological advances in both password creation and brute-force cracking. We support our findings by analyzing a sample of real-world, student passwords, and offer a look into the future of authentication.

Two previous studies have observed password selection habits and password strength of university students [9][10]. However, neither had made an attempt to compare the then-current passwords with older passwords examined in previous studies. In general,

comparison of past and present password security has been relatively scarce in literature, despite the oversaturation of the password security field. Feldmeier and Karn conducted a follow-up study ten years after the seminal paper by Morris and Thompson, focusing on the improvements in hardware and software [11]. Shen et al. compared the considered passwords with the passwords from several previous studies, though they did not include the study by Morris and Thompson [6]. More importantly, they did not evaluate the past and present passwords' resilience against different types of attacks, while taking user education and hardware improvements into account. Our work aims to close that gap.

3. Textual Passwords Through Time

The history of data breaches is as old as the idea of storing and protecting the digital data itself. Initially, incidents began to occur in companies that kept sensitive, business-oriented data in shared systems – one of the first reported dates back to the mid-1960s, when the CTSS operating system exposed all users' passwords as a daily welcome message [12]. In November 1978, the largest bank theft in U.S. history at the time occurred due to bad password practices [13].

Security officials and administrators of these systems were the first ones to warn about password security vulnerabilities. They suggested stronger and computationally slower encryption algorithms, salted passwords, and encouraged the users to create more complex passwords [1]. Nonetheless, reports of breaches from the early days of computing come mostly in the form of anecdotal knowledge – public awareness of password security threats did not begin to rise until the early 1990s, when the development and expansion of the Internet began to lead to the data explosion and reliance on Information Technology we are witnessing today.

While records of data breaches exist from that time [14], the majority of the largest data breaches have occurred and, consequently, been reported, in the last decade. One of the first most notable leaks in the recent years happened in 2009, when hackers managed to obtain more than 32 million plaintext passwords from the gaming website RockYou. In 2011 alone, over 70 million unencrypted passwords were leaked from various Chinese websites. It should be noted that such publicly known data breaches affect only a small margin of password-protected systems online, meaning that a much larger number of stored unencrypted passwords likely still exists in the wild.

Worryingly, even hashing does not warrant full safety of the stored passwords. Upon gaining access to a database of hashed passwords, an adversary only needs to determine the hash function used before she can attempt an offline brute-force attack. With continuous advances in processing speeds, it is becoming increasingly easier to find matching hashes. One of the persistent problems is that many systems continue to use deprecated, fast hashing algorithms. For example, around 2 million Ubuntu forum users' e-mails and MD5-hashed passwords were compromised in a 2013 hack. In 2016, the business-oriented social network service LinkedIn released a statement in which they acknowledged that a large set of hashed passwords was being sold on the black market. It turned out that at least 117 million passwords must have been leaked in 2012, of which 85% of SHA-1 hashes had already been cracked. Despite SHA-1 being

announced deprecated in 2011, industry remains too slow to move to safer alternatives. Variably iterative and memory-hard hashing algorithms were also proposed to render a brute-force attack too time-consuming to attempt. Nonetheless, even this solution knows examples of incidents. In 2016, over 43 million passwords were stolen from the web design platform Weebly, despite having been protected by bcrypt.

Security breaches have affected even large and well-known companies such as Adobe, Dropbox, eBay, MySpace, Tumblr, and Yahoo, to name a few.[1] The last is also an example of the extent of damage such incidents can cause: Yahoo reported that an overwhelming 1.5 billion user account` credentials had been stolen in hacking incidents that occurred in 2014 and 2015. The main reason for the sudden increase in the frequency and severity of these attacks stems mainly from the exponential growth of data processed online, which offers the potential attackers a realistic opportunity to expose large amounts of sensitive data by targeting a single online service. Additionally, developing technologies, such as IoT, pose many yet unresolved challenges that can be exploited [16].

It is safe to assume, and even expect, that this upward trend will continue in the future. Even more concerning is the prospect of such attacks becoming increasingly successful with the advances in technology. Hardware improvements have mostly been in accordance with Moore's Law ever since its establishment more than 50 years ago. While continuous exponential growth of processing power has already begun to slow down due to physical limitations, most semiconductor forecasters predict that Moore's Law will remain relevant for at least another ten years. By that point, factors other than transistor size might prolong the longevity of the projection: Beyond Intel's tri-gate transistors, we could be expecting monolithic 3D chips in the next decade, which would allow multiple layers of components built up vertically on a single silicon die. Alternative materials, such as indium or arsenide alloys, as well as different forms of carbon such as graphene or nanotubes, are promising as well, due to their highly conductive properties and lower power consumption.

Farther down the line, we might be looking at the more radical successors of the silicon era. One possibility is spintronic semiconductors, which exploit electrons' rotational energy rather than their charge to represent bits. While this technology has been in development for more than 15 years, there are still challenges that need to be overcome before it can reach the production stage [17].

Quantum computing is another revolutionary approach that has the potential to mark one of the most impactful leaps in computer history since the establishment of the Von Neumann architecture. Due to the nature of their operation, quantum computers excel at particular types of mathematical tasks. Prime factorization and discrete logarithm problems are two types of hard mathematical problems, used widely in cryptography, that could be solved in polynomial time using Shor's algorithm [18]. As fundamental questions such as what to build qubits out of and how to maintain quantum superposition are answered, we might be looking at drastic increases in processing power, equivalent to several orders of magnitude, which would make password breaking a trivial task.

Despite the aforementioned and other possible future improvements in hardware that are still in development, there are other ways for potential adversaries to speed up

[1] InformationIsBeautiful collects data about recent data breaches based on media reports, illustrating the growing trend through meaningful visualization [15].

password cracking times. The benefit of parallel processing over sequential searches allows modern GPUs to offer more than 10-fold increases in processing speeds when compared to general processors, making them a viable choice for executing brute-force attacks; increasing the number of nodes in the cluster provides linear scalability with very little to no overhead. Dedicated FPGA cores can also be employed to improve password guessing speeds when specialized password-hashing functions (i.e. bcrypt or scrypt) are used to slow down the computation time intentionally [19]. Recently, more advanced techniques that attempt to optimize the password guessing order have begun to emerge. Weir et al. suggested the use of probabilistic context-free grammars, which can be derived from training sets of previously revealed passwords. These grammars provide a basis to generate word-mangling rules, which determine the guessing order of individual passwords, based on the estimated probability of their occurrence [20]. Exploiting the distribution of letters within the passwords, Markov chain models work on the assumption that most users choose easy-to-remember passwords that follow the rules of their native language. Unlikely passwords are, thus, removed to reduce the initial search space, while the remaining passwords can be ordered by the probability of a given password's characters appearing in a specific order. This approach provides a substantial improvement over the classic, straightforward dictionary attack [21].

The growing frequency and effectiveness of security breaches cannot be attributed only to computing advances and improvements in cracking techniques, however. Passwords can also be retrieved by exploiting system vulnerabilities, such as the COUPLE attack [22]. According to [15], some of the largest leaks happened because passwords were lost, stolen, or published accidentally. Furthermore, numerous studies have observed users' password creation and management practices, and have concluded that humans are the weakest link in password security [23][24][25]. Their password selection methods are influenced by the characteristics of the human memory. First, cognitive burden increases with the number of elements that a user is required to memorize. Second, humans are decisively better at remembering meaningful strings they can recall based on associations rather than random sequences. Due to these limitations, humans generally gravitate toward generating short, simple, and predictable passwords. While such passwords might be easy to remember for humans, they are, unfortunately, also easy to guess for machines: Short length and low password complexity allows them to traverse the entire available search space and 'find' the correct password in a very short time.

On the other hand, forcing the users to authenticate using a computer-generated, random password might not be the optimal solution. A study on the memorability of high-entropy passwords has shown that the longer the character string is, the more difficult it is for humans to remember and recall it later. At the same time, it takes longer for them to input the password and they are more likely to make an error [26].

One of the primary challenges in password security has, thus, been finding the balance between security and memorability. Password policies are one of the most well-known and established approaches to reach the middle ground. They were introduced with the intention of ensuring a sufficient level of security while still allowing the users to retain some originality when creating their own passwords. As expected, stricter password policies generally provide a higher level of security, at the expense of user convenience. Lee and Choong recommend simple rule sets, arguing that significant drops in usability do not outweigh the minimal increases in security provided by the

more complex rule sets [27]. Nonetheless, studies suggest that the relationship between security and perceived usability by the users is not dependent only on the strictness of password policies [28]. Well-crafted policies are supposed to offer an equal or higher degree of security, with slight, or even insignificant, decreases in usability, when compared to sub-par policies. However, there are no absolute guidelines to creating good password policies; most authors agree that optimal policies are difficult to create [27][28]. Furthermore, poorly formed policies that do not take into account users' psychological profiles, their work practices, and perceived importance of their accounts, can actually be more damaging than beneficial [29]. The inclusion of the surrounding context is recommended not only for textual passwords, but access control models as a whole [30]. In practice, policies often impose high constraints for limited benefit; Bonneau et al. suggests they might not actually reduce harm [8].

Users' lack of security knowledge is yet another problem that has been largely pointed out in scientific literature. It has been suggested that, in order to improve password composition and management practices, it is crucial to raise awareness of security-related threats and educate the users [29][31]. However, recent studies show that users know what constitutes a good password and are well aware of the consequences of bad password mismanagement [23], but are willing to trade security for comfort [31]. Consequently, they exhibit lax security behavior: They are disinterested and frustrated with password policies, and resort to bad password management practices out of laziness and convenience [27]. In order not to forget their passwords, users avoid changing them [32] or write them down [31][32]. If they are forced to change their passwords upon expiry, they tend to select similar passwords [33]. To reduce the memory load, they are also known to reuse their passwords across several services [32][34][35], or share them with their family members or friends [31][34].

The extent of these bad practices reveals that it is not enough for the users only to be aware of the importance of password security – they must be motivated to protect their personal data. Because of that, they are more likely to use stronger passwords for accounts they perceive to be of high importance (e.g. bank accounts) [25][35], though the actual strength of the chosen passwords still depends on their understanding of what constitutes a good password [32]. Furthermore, their attitude towards security-related issues is influenced by their past experiences. If they are using their accounts continuously without any security incidents affecting them, users feel less threatened, or even begin to doubt that their accounts can be breached. Studies suggest that, in order for them to become security-conscious, they need to feel a personal loss as a result of their own data being compromised [24][32].

4. Comparison of Past and Present

37 years ago, Morris and Thompson conducted a study in which they collected and analyzed 3,289 user-generated passwords. They divided them into several categories based on their length and composition and found that 2,831 (or 86.07%) of those passwords could be recovered in a matter of days. Finally, they provided several suggestions on how to improve the system in order to prevent an adversary from being able to execute a brute-force attack in a reasonable amount of time [1]. Several of their

propositions (such as improved hashing functions, salted passwords, and password policies) later became a widely-used means to combat the ever-growing computer processing power.

Over the next few decades, many studies observed the characteristics of users' passwords leaked at the time, and pointed out that they are alarmingly bad. However, to the best of our knowledge, none of them provided a direct comparison of current and past password strength while also taking into consideration advances in technology, user education, and security procedures (e.g. password policies). In the current era of exponentially growing data, and the corresponding increase in security-related incidents, it is essential to ask ourselves whether there have been any improvements in the field of Password Security.

For the purposes of this study, we evaluated password strength based on the amount of time it would take to try all possible passwords of a given length and composition. In the original study by Morris and Thompson, it was estimated that it would take a second to check about 800 encrypted passwords on a UNIBUS system PDP-11/70, containing what was, at the time, a high-performance CPU [1]. To make a valid comparison, we chose a contemporary high-end processing unit. In a recent article, Qui et al. measured performance of AMD GPUs generating MD5 and SHA-1 password hashes. After optimizing the hashing algorithm and password generation, they were capable of achieving the speed of 6,877 million and 2,615 million tries per second for MD5 and SHA-1 respectively, on a high-end desktop graphic card AMD HD7970 [36]. Additionally, we compared the cracking times of an 8 nVidia GeForce GTX 1080 Ti rig generating much slower bcrypt hashes. According to the latest benchmark results, the rig achieved up to 185,000 tries per second for bcrypt running 5 iterations (32 rounds) [37].

We calculated the running times for an AMD HD7970 unit generating SHA-1 hashed passwords, as well as a rig of eight nVidia GeForce GTX 1080 Ti units generating bcrypt hashes. Bcrypt's adjustable work factor allowed us to explore the running times for three configurations: Low (5), medium (12), and high (20) numbers of iterations. The choices of work factors are not arbitrary: Five iterations reproduce the setup used in all benchmark tests, twelve iterations were chosen because they are often used in practice and currently accepted as the security standard, while twenty iterations represent a high-end alternative for security-sensitive applications.

A direct comparison of SHA-1 running times with the running times reported by Morris and Thompson revealed that there has been a substantial decrease (of over 326 million percent) in the amount of time it takes for a state-of-the-art computer to traverse the same given search space of available passwords. That change is roughly in line with Moore's law; faster computation was balanced by slower and more complex hashing algorithms that were developed over the years. Hash generation times indicate that much longer SHA-1 passwords can be retrieved nowadays as opposed to encrypted passwords 37 years ago. A potential adversary with access to a commercially available AMD HD7970 could easily test strings up to 7 characters or lowercase alphanumeric passwords up to 9 characters in length. By utilizing more powerful processing units and then distributing the task of password hashing across several of them, it would be possible to find even longer and more complex passwords.

However, recent improvements in password hashing (such as Argon2, bcrypt and scrypt) once again limit a potential adversary's endeavors. These iterative, memory-hard functions are designed to slow down password hashing, even with constant increases in

computer processing power. When comparing the running times of a single GPU AMD HD7970 generating SHA-1 hashes with eight nVidia GeForce GTX 1080 Ti computing bcrypt hashes with work factor 5, we found the latter were computed more than 14 times slower, despite the much larger processing power dedicated to the password hashing task. Adding just one additional iteration already doubles the password hashing time. While such hashing speeds are by no means slow enough to protect against brute-force attacks, they illustrate the potential of memory-hard functions: As computer processing power continues to increase, security experts will merely have to increment the work factor to compensate for the decreased time necessary to perform simple operations when computing password hashes.

To determine the hashing algorithms that are resistant to offline brute-force attacks, we devised two scenarios. In both scenarios, the attacker has obtained a full list of hashed passwords, and decides to perform a brute-force attack to uncover passwords. The difference between the scenarios is in the attacker's objective. In the first scenario, the adversary simply wants to get access to the system, meaning that she will be trying multiple passwords one after another, expecting to find a particularly weak password that can be cracked in a relatively short time. For that purpose, we chose one full day as a cut-off point for cracking a single password before the attacker moves onto the next one. In the second scenario, the attacker is targeting a particular user's password, and is willing to direct all available resources to uncover it. We assume, however, that she can spend up to a year trying to crack the password, before giving up under the assumption that the said password cannot be recovered in a reasonable amount of time.

Table 1. Password length and estimated cracking times for two hashing algorithms: SHA-1 ran on an AMD HD7970 GPU, and bcrypt with low (5), medium (12), and high (20) work factor, ran on 8x nVidia GTX 1080 Ti rig

	pool size	10	26	36	52	62	95	128
cutoff point	algorithm	digits	lowercase	lowercase & digits	alphabetic	alphanumeric	printable ASCII	all ASCII
One day	SHA-1	14 chars (10h 37m)	10 chars (15h)	9 chars (10h 47m)	8 chars (5h 41m)	8 chars (23h 12m)	7 chars (7h 25m)	6 chars (28m 1s)
	bcrypt (5)	10 chars (15h 2m)	7 chars (12h 4m)	6 chars (3h 16m)	5 chars (34m 17s)	5 chars (1h 23m)	5 chars (11h 38m)	4 chars (24m 12s)
	bcrypt (12)	8 chars (19h 14m)	5 chars (2h 17m)	5 chars (11h 38m)	4 chars (1h 24m)	4 chars (2h 51m)	4 chars (15h 40m)	3 chars (24m 12s)
	bcrypt (20)	5 chars (4h 56m)	4 chars (22h 30m)	3 chars (2h 18m)	3 chars (6h 56m)	3 chars (11h 44m)	2 chars (26m 40s)	2 chars (48m 25s)
One year	SHA-1	17 chars (1.21Y)	12 chars (1.16Y)	11 chars (1.59Y)	10 chars (1.75Y)	10 chars (10.17Y)	9 chars (7.64Y)	9 chars (111.77Y)
	bcrypt (5)	13 chars (1.71Y)	10 chars (24.21Y)	9 chars (17.42Y)	8 chars (9.17Y)	8 chars (37.44Y)	7 chars (11.97Y)	7 chars (96.53Y)
	bcrypt (12)	11 chars (2.19Y)	8 chars (4.58Y)	7 chars (1.72Y)	7 chars (22.56Y)	6 chars (1.25Y)	6 chars (16.13Y)	6 chars (96.51Y)
	bcrypt (20)	9 chars (5.62Y)	6 chars (1.74Y)	6 chars (12.23Y)	5 chars (2.14Y)	5 chars (5.15Y)	5 chars (43.48Y)	4 chars (1.51Y)

Table 1 summarizes the estimated cracking times for passwords hashed by either SHA-1 or bcrypt. In each cell, we list password length, and the amount of time it would take to crack that password. In particular, we are interested in cracking times for both

listed scenarios. In the first scenario, we are looking for the longest passwords of various compositions that can be cracked in less than a day. For example, any SHA-1 hashed alphanumeric password of 8 characters or less could be cracked in more than a day (an 8 character long password would take approximately 23 hours and 12 minutes). That allows us to identify passwords that are considered too weak to be protected by a given hashing algorithm. Ideally, we want none of the stored passwords to fall into that category. If there are some, or even the majority of passwords of a given composition that are of the same, or shorter length than the ones listed for a specific hashing algorithm, we encourage system administrators strongly to hash all of their passwords with a slower hashing algorithm.

The second scenario lists the lengths of the shortest passwords of several compositions that cannot be cracked in less than a year. For instance, a 13-digit password hashed with bcrypt (work factor 5) would take more than a year (1.71 years) to crack. We characterize passwords of that length and longer as strong; if most of the passwords of a given composition can be classified into that category, we can assume that the hashing algorithm used to protect these passwords is sufficiently secure. Naturally, with further advances in computer processing speeds, cracking times should be re-evaluated to determine whether the passwords should be re-hashed with a stronger hashing algorithm.

Aside from advances in computing, we should also consider decades of user education and enforced security procedures, which were employed primarily to increase security and slow down brute-force attacks. Back in the 1970s, at the time of Morris and Thompson's study, users were still unaware of the relevance of their role in protecting their own accounts. Nowadays, people are mostly aware of good password creation practices, resulting in them making longer and more complex passwords; many web sites also limit their bad password choices by enforcing password policies. Consequently, it would take longer to crack these passwords on the same machines, which affects the choice of hashing algorithms.

To determine the average present-day password strengths, we collected a total of 7,408 university passwords. These passwords were generated by the users between the years 2008 and 2014, and were used to protect real, sensitive data, such as students' personal information, e-mails, course materials and grades. An adversary gaining access to the accounts protected by these passwords could access any Wi-Fi network within the Eduroam system, download software and licenses free of charge, register the account owner for an exam, or cancel an existing exam registration.

The plaintext passwords were obtained from the university under a strict security policy and as a result of a several months long negotiating process. A computer unit with restricted physical access was devoted exclusively to storing and processing the data. The unit's web access was physically disabled at all times, and the operating system was protected by a username and password. The password data stored on the device was encrypted; the processing and statistical analysis was performed on temporarily unencrypted data, which was erased after processing.

As we were operating with real data, ethical concerns were addressed as well. The obtained data was anonymized by the university security service's personnel, and contained only plaintext passwords. The use of such personal data without prior or written consent of a subject (e.g. student) is allowed for the purpose of research under Article 11 (2), Article 13 (2), and Article 32 (3) of the Data Protection Directive [38].

Establishing a link between a student and their password would require access to their personal information (such as name, surname or student ID), which is kept in the university's databases; however, if a malicious party gained access to the database through any means, reverse engineering the obtained passwords would be unnecessary as the records could be accessed directly. Furthermore, we believe other factors assist in reducing a potential adversary's chance of an unauthorized breach of security and/or privacy to a minimum. The majority of passwords belong to students that no longer use them (graduates or drop-outs), and whose accounts are thus no longer active.

Our main objective was to determine how vulnerable individual passwords are to current password cracking. For that purpose, we considered the first cracking scenario, in which the adversary would attempt to crack each password for a full day before moving onto the next one. Similarly to Morris and Thompson, we modeled a two-stage attack: In the first part, we estimated the amount of time it would take to find a given password in an exhaustive brute-force attack, and in the second part, we checked the passwords against various dictionaries.

Due to time and processing power constraints, we did not execute the brute-force attack on all passwords. Instead, each password was classified into one of several categories based on their length and composition. In the original article by Morris and Thompson, the categories included passwords that could be retrieved in a matter of days on a PDP-11/70. For comparison, we compiled a new list of categories for every considered hashing algorithm, to benefit current-age security officers. Each individual password that was classified into one of these categories could be retrieved in less than a day on modern-age GPUs.

Next, we performed a dictionary attack on the collection of SHA-1 hashed passwords, using several available wordlists. Altogether, we recovered 668 distinct passwords (approximately 9% of the set), however, only 47 passwords have not already been found in the brute-force attack. Such a simple attack could be executed in just ten minutes; by including diverse wordlists, and applying advanced techniques such as word mangling rules, we believe a more substantial percent of passwords could have been recovered.

In total, each out of 5,992 SHA-1 hashed passwords or 81% of the analyzed collection could be found in less than a day on a modern high-end graphics card using a simple brute-force and dictionary attack. This figure is alarmingly close to the 86% of user-generated passwords uncovered by Thompson and Morris back in 1979. Even if the average password length and complexity has increased over the years, the change is too slight to counter the rapid advances in processing power. Considering the vast number of websites that still use the SHA-1, the deprecated MD5, or even save their passwords in plaintext, we can conclude that that there has not been much improvement in password security in the past three and a half decades.

On the other hand, should the websites move to stronger hashing algorithms, such as bcrypt, the amount of recovered passwords would decrease dramatically. In our case, simply hashing all passwords with a bcrypt algorithm with the work factor 5 would mean that less than half (47.8%) of the passwords could be recovered in less than a day on a powerful GPU rig. Increasing the work factor to the conventional 12 would mean only about 3.6% less passwords could be cracked, though a further breakdown indicates why that is the case; while 501 (6.76%) less passwords could be recovered in a brute-force attack as a result of the higher work factor, 236 (3.19%) more passwords were instead found in dictionaries. Most of these passwords were simple, lowercase passwords of 6

and 7 characters in length. They illustrate that even a fortified defense against brute-force attacks does not yet guarantee that the passwords cannot be compromised via another attack channel; all of these passwords are still vulnerable to dictionary, social engineering, and other types of attacks on passwords.

Table 2. Number of passwords that could be cracked in less than a day for a given hashing algorithm, based on password length and composition

	Morris & Thompson (1979)	Student passwords (2008-2014) SHA-1	bcrypt (5)	bcrypt (12)	bcrypt (20)
1: Brute-force	2339 (71.11%)	5945 (80.25%)	3194 (43.12%)	2693 (36.35%)	97 (1.31%)
ASCII	551 (16.8%) 1-3 chars	3305 (44.6%) 1-7 chars	190 (2.6%) 1-5 chars	94 (1.3%) 1-4 chars	2 (0.03%) 1-2 chars
Alphanumeric	477 (14.5%) 4 chars	1737 (23.4%) 8 chars			3 (0.04%) 3 chars
Alphabetic	706 (21.5%) 5 chars				
Lowercase & digits		763 (10.2%) 9 chars	2751 (37.1%) 6 chars	96 (1.3%) 5 chars	
Lowercase	605 (18.4%) 6 chars	132 (1.8%) 10 chars	163 (2.2%) 7 chars		86 (1.2%) 4 chars
Digits		8 (0.1%) 11-14 chars	90 (1.2%) 8-10 chars	2503 (33.8%) 6-8 chars	6 (0.08%) 5 chars
2: Dictionary	492 (14.96%)	47 (0.63%)	347 (4.68%)	583 (7.87%)	638 (8.61%)
Total	2831 (86.07%)	5992 (80.89%)	3541 (47.80%)	3276 (44.22%)	735 (9.92%)

Nonetheless, bcrypt with work factor 12 still allowed for any out of more than a third of user-generated passwords to be recovered in less than a day. The majority of these were digits of 6 to 8 characters in length; most of the alphabetic and alphanumeric passwords were long enough (above 5 characters) to withstand a day-long brute-force attack. Even so, if a large portion of the password set is vulnerable to brute-force attacks, administrators are encouraged to protect the passwords with a stronger hashing algorithm. We considered the bcrypt algorithm with work factor 20, which would greatly increase the resilience against brute-force attacks, at the expense of performance. Only 97 out of 7,408 passwords (a little over 1%) could potentially be recovered in less than a day, though a simple dictionary attack could find 638 more passwords (all were lowercase passwords of 5 characters and more), bringing the total to 735, or nearly 10% of the password set.

5. Advice to the Partakers

Much of the blame for the password security problem over the past decades has been placed on the shoulders of the users. Extensive research has focused on mitigating the issue from the user viewpoint, but, at the end of the day, even if the users can be educated to adopt good password creation and management habits, the human memory capacity will remain limited. As technology continues to progress, it is unreasonable to expect humans will be able to follow such a pace, even if they are familiar with the often clashing, limited-benefit, or even counter-productive requirements, they will eventually become unable to fulfil them.

That does not, however, imply that the users do not have to be aware of the password-related problems. While it should not be on them to devise passwords capable of sustaining prolonged and dedicated brute-force attacks, their efforts when choosing and managing their own passwords can greatly diminish the effectiveness of some other types of attacks, particularly social engineering. Actively avoiding the choice of words, conventional symbol replacements, and personal information in passwords are still good password practices to adhere to, but users remain slow to adopt them because they have no motivation and see no immediate gain. Rather than giving out advice (which is often ignored), or enforcing minimum requirements (which can lead to frustration), we believe users should be stimulated by positive feedback (such as visual indicators of password strength), and user choice. In particular, a system could provide several suggestions to improve a password during the account creation, allowing the user to choose the most preferred one. Similarly, a system could encourage the user to incorporate personal associations into their passwords by increasing the score when password choice includes syllables derived from multiple words, combined with symbols.

At the same time, as more users are persuaded to trade convenience for greater security (e.g. password managers, or two-factor authentication), we are working towards a global shift in the way of thinking. Just like users have shifted gradually towards longer and more complex passwords over the last few decades as a result of education, once a certain threshold of security-conscious users is reached, prioritizing security over convenient use should become the norm, rather than an exception.

While continuous and constant user education would certainly help, we identify it as only a part of, rather than a wholesome solution, to the long-lasting password security problem. Proportionally, a much larger responsibility lies on the shoulders of security officers and administrators: While users need only to manage their personal accounts, administrators should ensure the security of the entire system. Given that many users are not expected to realize the security requirements for their own accounts, administrators are required to implement sufficient security measures to protect all accounts against a wide variety of possible attacks, ranging from brute-force and dictionary, to traffic interception and system vulnerability exploitation [22]. Since many of these accounts and their passwords might be particularly poor, it is all the more important for the administrators to ensure a high level of security. Unfortunately, there are no all-encompassing rules or guidelines that all administrators should conform to. The appropriate level of security is unique to the system, and largely dependent on the profile of a typical user (e.g. customers vs. CEOs), sensitivity of stored data (e.g. consumer data vs. healthcare data), as well as the scope and purpose of the system (e.g. small retail systems vs. social networks). Security officials are strongly advised to assess security requirements based on these variables and then design, implement, and continuously review a comprehensive security policy tailored to the system in question.

Current approaches to authentication security remain simple and often ad-hoc. Most services attempt to achieve a sufficient level of security by enforcing password policies on the users. However, too few of them stop to consider exactly which attacks they are defending against and what the implemented policies are trying to accomplish. Often, that leads to administrators implementing strict password policies which are attempting to protect the accounts from several independent attacks. However, as a result, they limit users' password choices greatly, which can, in turn, affect their motivation, and cause them to fall back into bad password habits. At the same time, overly strict policies might

actually benefit the attacker – since she is aware of the requirements, and can reasonably suspect many users will try only just barely to match them, which allows her to tailor her guesses in a way to check the most likely passwords first. Instead, administrators should consider using password policies as a way to mitigate one, or perhaps just a few targeted attacks. Dictionary attack comes up as the most viable option, as it prevents the users from choosing simple, and predictable passwords that could otherwise withstand a brute-force attack due to their lengths.

At the same time, different measures should be implemented to safeguard the passwords against other types of attacks. Already Morris and Thompson had suggested for stored passwords to be hashed and salted as a countermeasure to brute-force attacks. In our experiment, we have shown that hashing passwords with deprecated algorithms can be as inefficient as storing them in plaintext. Administrators are, thus, strongly urged to keep updated on the improvements in password hashing. A good approach is to evaluate the passwords as they are created, then choose a hashing algorithm based on the general strength of the passwords stored. If the weakest passwords in the database could be cracked in a reasonable amount of time (i.e. less than a day), administrators should consider re-hashing all of the passwords with a stronger hashing algorithm. The general idea is not to wait until the hashing function in use becomes obsolete, but instead always remain a step ahead.

6. The Future of Authentication

The argument that improvements in password cracking imply the end of passwords, has been made many times over the past few decades. For example, St. Clair et al. predicted password exhaustion based on the prevailing and predicted computer processing power, shown to be capable of cracking most real passwords used by the university students [39]. Various tabulations of cracking speeds have been offered repeatedly either to cajole users into stronger password choices, or convince them that passwords must be replaced. In the last decade especially, numerous parties advocating for alternatives to passwords made the claims that the passwords are dead.

However, it remains to be answered whether deprecating textual passwords and moving onto another authentication method entirely, is really the best answer. Research on authentication methods is bloated with proposals on novel authentication schemes aimed at replacing textual passwords. Many promote their improvements over passwords, but then fail to address weaknesses that often render them unusable in practice. Bonneau et al. examine closely many diverse authentication methods proposed in recent years from several angles, showing that all of the suggested alternatives suffer from serious drawbacks that make them less suitable than textual passwords for many purposes [40]. To name just a few:

- Graphical passwords may increase memorability because of their visual aspect, but suffer from most of the problems that their textual counterparts do, such as guessing (brute-force, dictionary) and capture (malware, phishing, social engineering) attacks, and a few more besides. Most prominently, the increased security of these schemes generally means decreased usability, [27]

- Hardware tokens provide higher security while reducing the need to memorize additional information (with the exception of the verifier PIN). However, they also create a significant inconvenience for the user, as she cannot authenticate without the token on her person. A stolen or lost token would also need to be revoked and re-issued. Because of that, it is unlikely to expect that users would want to use tokens for a multitude of everyday accounts in everyday situations,
- Biometry allows for an easy-to-use and compelling way of authentication, while eliminating the need for the user to memorize information or safeguard physical tokens. For the time being, the high cost and poor accuracy of this technology remains a problem. Even if there is room for improvement in both cases, biometric solutions still have poor deployability when compared to passwords. By far the most concerning problem, however, comes into play with a possible breach of the database of secrets. While biometric credentials can be stolen, extorted or forged relatively easily, they cannot be revoked without blocking both malicious and legitimate use in the process,
- Cognitive schemes share several security and usability benefits with textual passwords, even if it would be generous to say they are on par. For example, while textual passwords can be typed from muscle memory, cognitive passwords are input slower because they require effort and attention. Furthermore, minimizing secret leakage increases cognitive workload, resulting in longer login times and higher login errors,
- Password managers, while offering advantages over classical passwords in terms of security and selected usability aspects, lack in terms of deployability (accessing the database of secrets from another device requires Internet connection and a proprietary application; alternatively, it can be stored on a physical device, but it can be lost or stolen) which affects convenience of use. Most notably, however, they still rely on the underlying technology of textual passwords. Even so, Bonneau et al. argue they could become a common coping strategy, should the passwords remain widespread. [26]

To understand the complexity of the age-old password security dilemma, we can define it as a multi-objective optimization problem. In this context, security is just one of several contradicting parameters that we are trying to optimize. If we visualize existing authentication methods as points in a multi-dimensional space in which axes represent the parameters, we are essentially looking for methods that have one, or preferably several, parameters maxed out (i.e. the Pareto front).

Naturally, researchers should be encouraged to keep investigating new authentication possibilities, while striving to improve the already existing ones. The main goal of the former is to search the security-usability-deployability continuum thoroughly for any authentication methods that appear on or beyond the Pareto front. The latter aims to push the existing schemes closer toward, and hopefully past, the current front. Both research directions are important for us to gain a comprehensive understanding of the field, so that a more educated approach toward better authentication can be taken in the future.

The current predicament is that the researchers do not take a step back to assess objectively what has already been discovered and try to fit their own piece into the authentication mosaic. Instead, they work under the assumption that the existing research on authentication mechanisms is conclusive. They often assume a biased view of the

measures that highlight their method's strengths, and are focused primarily on trying to advocate for their new or improved methods to either complement or replace textual passwords. Such approach can distort academia's view of the authentication problem, further enforcing the idea that a shift in the field of Authentication is necessary and imminent.

However, our heavy reliance on textual passwords stands in stark disagreement with this argument. Were any of the alternatives scoring better than textual passwords in several competing aspects, they would likely have replaced them over the years. Still, at least for the time being, textual passwords undoubtedly remain on the Pareto front of authentication mechanisms. The users have no incentive to switch them for any other alternative on the front, as that would effectively be trading one set of advantages for another.

It is largely true that user-chosen passwords cannot withstand brute-force attacks. However, mechanisms such as salting and strong, up-to-date hashing algorithms, can diminish the effectiveness of these attacks. The means to defend against this weakness exist – it is on the shoulders of system administrators to ensure the safety of users' accounts from this point of view. It is idealistic to expect that all of them will do so; consequently, we are likely to keep hearing about new security breaches in the future. However, in the grand scheme of things, these are more or less isolated cases, and researchers suggest that offline attacks (which are most often studied in password literature) are not as common as we might think [8].

It is particularly difficult to assess textual passwords' susceptibility to other types of attack in real-world scenarios. When examining the world's largest data breaches in the last decade, most of the passwords were either exposed in accidental or deliberate leaks, or obtained from information system hacks and, subsequently, cracked [15]. Successful cracking (where necessary) was made possible due to the deprecated and/or insecure hashing algorithms used to protect them. That further supports the claim that the way passwords are stored matters quite a bit, as Florencio et al. discuss [42]. Regardless, password guessing has moved beyond targeted brute-force and dictionary attacks. For example, social engineering techniques can be used to take advantage of the human factor in password creation. The current state-of-the-art cracking employs recurrent neural networks, which provide a much more accurate sense of password vulnerability [43]. Further research should examine how feasible these attacks are in real-world scenarios, and attempt to assess how often they actually appear in practice.

When predicting textual passwords' longevity and the future of authentication, we believe two points should be considered. The first is the users' motivation to switch to another authentication scheme. Despite reported attacks and breaches of password-protected accounts, as well as constant cajoling by both stakeholders and the media that the passwords should be replaced, users are not compelled to replace them. Most are convinced a breach of privacy cannot happen to them, and will not have a shift in their way of thinking unless they experience personal loss. Therefore, the very small percentage of users affected by data breaches is nowhere near the threshold necessary for the entire population to consider switching textual passwords for another authentication method. Their motivation is also affected partially by the lack of competitive alternatives. That highlights the second problem: Users are unlikely to favor another authentication method if it would mean nothing but trading different sets of

advantages. Until there are no decisively better alternatives for the users to shift towards, textual passwords are likely to remain in widespread use.

7. Conclusion

Already in 1979, Morris and Thompson inspected a set of real, user-generated passwords, and concluded that they were inadequate. In an endeavor to increase its security, they suggested several improvements to the original authentication scheme. More than three and a half decades later, we are witnessing an upsurge in data breaches, while the security officers contemplate whether to replace textual passwords or not.

In this article, we make several important contributions to the resolution of this dilemma and the field of Password Security as a whole. First, we make a direct comparison of recent, real-world passwords and the passwords collected by Morris and Thompson. Through historical analysis, we show that there has been no efficient change in password security in the last couple of decades. The main reason why that is the case stems from the fact that neither users nor system administrators conform to Morris and Thompson's advice diligently. As a result, textual passwords are more vulnerable to the attackers, prompting more individuals from academia and the industry alike to advocate for a shift in security.

Our second contribution is, therefore, an advice to administrators to implement password checkers that will forbid the use of dictionary words in textual passwords. In our article, we establish that the composite of technological solutions, security procedures, and user education, as proposed by Thompson and Morris, can still be effective even today. We demonstrate that employing memory-hard hash functions like bcrypt can protect passwords despite technological advancements – but against brute-force attacks only. Nothing prevents users from choosing simple, predictable passwords that are vulnerable to dictionary attacks. Nearly forty years have taught us that users will not change. It is, therefore, vital for administrators to realize that fact and act upon it. It will be easier to change the administrators' behavior than the users'.

Finally, we believe that the currently available alternatives to passwords cannot solve the security problem. Their few advantages over textual passwords often come with an extensive list of disadvantages that are poorly documented and validated in literature. Currently, no existing solution outperforms textual passwords in terms of security, usability and deployability. Ultimately, we do not expect textual passwords will be replaced in the foreseeable future. While an imperfect form of authentication, their shortcomings will be overlooked as long as they are considered "good enough" by the typical user.

In this article, we assessed the current state of textual passwords critically and provided insight on how password security can be improved from the perspective of both users and system administrators. We believe such approaches could reinforce their position as the dominant authentication method in the future. Regardless, we do not dismiss the existence of a better authentication scheme. Thus, we urge researchers to keep developing new and improving existing authentication schemes in an effort to expand our understanding of authentication security and identify Pareto-improving authentication methods.

Acknowledgement. The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0057).

References

1. Morris R., Thompson K.: Password Security: A Case History. Communications of the ACM, Vol. 22, No. 11, 594–597. (1979)
2. Zviran M., Haga W. J.: Password Security: An Empirical Study. Journal of Management Information Systems, Vol. 15, No. 4, 161–184. (1999)
3. Florencio D., Herley C.: A Large-Scale Study of Web Password Habits. In Proceedings of the 16th International Conference on World Wide Web. Banff, AB, Canada, 657–666. (2007)
4. Dell'Amico M., Michiardi P., Roudier Y.: Password Strength: An Empirical Analysis. In Proceedings of the 29th IEEE International Conference on Computer Communications. San Diego, CA, USA, 983–991. (2010)
5. Bonneau J.: The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In Proceedings of the 33rd IEEE Symposium on Security & Privacy, San Francisco, CA, USA, 538–552. (2012)
6. Shen C., Yu T., Xu H., Yang G., Guan X.: User Practice in Password Security: An Empirical Study of Real-life Passwords in the Wild. Computers & Security, Vol. 61, 130–141. (2016)
7. Ji S., Yang S., Hu X., Han W., Li Z., Beyah R.: Zero-Sum Password Cracking Game: A Large-Scale Empirical Study on the Crackability, Correlation, and Security of Passwords. IEEE Transactions on Dependable and Secure Computing, Vol. 14, No. 5, 550–564. (2017)
8. Bonneau J., Herley C., van Oorschot P. C., Stajano F.: Passwords and the Evolution of Imperfect Authentication. Communications of the ACM, Vol. 58, No. 7, 78–87. (2015)
9. Mazurek M. L. et al.: Measuring Password Guessability for an Entire University. In Proceedings of the 13th Conference on Computer & Communications Security, Berlin, Germany, 173–186. (2013)
10. Awad M., Al-Qudah Z., Idwan S., Jallad A. H.: Password Security: Password Behavior Analysis at a Small University. In Proceedings of the 5th International Conference on Electronic Devices, Systems and Applications, Ras Al Khaimah, UAE. (2016)
11. Feldmeier D. C., Karn P. R.: UNIX Password Security - Ten Years Later. In Proceedings of the 9th Conference on the Theory and Applications of Cryptology. Santa Barbara, CA, USA, 44–63. (1989)
12. Corbató F. J.: On Building Systems That Will Fail. In: ACM Turing Award Lectures. ACM, New York, 72–81. (2007)
13. Mitnick K. D.: The Art of Deception – Controlling the Human Element of Security. John Wiley & Sons, Indianapolis, USA. (2002)
14. Hancock B.: Network Breaches: They are Real. Computer Fraud & Security, Vol. 1998, No. 10, 8–11. (1998)
15. McCandless D.: World's Biggest Data Breaches. (2018). [Online]. Available: http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/ (current July 2018). Archived by WaybackMachine® at http://web.archive.org/web/20180711024143/http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/
16. Mavropoulos O., Mouratidis H., Fish A., Panaousis E., Kalloniatis C.: A Conceptual Model to Support Security Analysis in the Internet of Things. Computer Science and Information Systems, Vol. 14, No. 2, 557–578. (2017)

17. Ranmohotti K. G. S. et al.: Coexistence of High-Tc Ferromagnetism and n-Type Electrical Conductivity in FeBi2Se4. Journal of the American Chemical Society, Vol. 137, No. 2, 691–698. (2015)
18. Shor P. W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing, Vol. 26, 1484–1509. (1997)
19. Wiemer F., Zimmermann R.: High-Speed Implementation of bCrypt Password Search using Special-Purpose Hardware. In Proceedings of the 2014 International Conference on Reconfigurable Computing and FPGAs. Cancun, Mexico. (2014)
20. Weir M., Aggarwal S., De Medeiros B., Glodek B.: Password Cracking using Probabilistic Context-Free Grammars. In Proceedings of the 30th IEEE Symposium on Security and Privacy. Berkeley, CA, USA, 391–405. (2009)
21. Narayanan A., Shmatikov V.: Fast Dictionary Attacks on Passwords using Time-Space Tradeoff. In Proceedings of the 12th ACM Conference on Compututer and Communications Security. Alexandria, VA, USA, 364–372. (2005)
22. Cho B., Lee S., Xu M., Ji S., Kim T., Kim J.: Prevention of Cross-update Privacy Leaks on Android. Computer Science and Information Systems, Vol. 15, No. 1, 111–137. (2018)
23. Tarwireyi P., Flowerday S., Bayaga A.: Information Security Competence Test with regards to Password Management. In Proceedings of the 2011 Conference on Information Security South Africa. Johannesburg, South Africa. (2011)
24. Tam L., Glassman M., Vandenwauver M.: The Psychology of Password Management: A Tradeoff between Security and Convenience. Journal of Behaviour & Information Technology, Vol. 29, No. 3, 233–244. (2010)
25. Notoatmodjo G., Thomborson C.: Passwords and Perceptions. In Proceedings of the 7th Australasian Conference on Information Security. Wellington, New Zealand, Vol. 98, 71–78. (2009)
26. Stanton B. C., Greene K. K.: Character Strings, Memory and Passwords: What a Recall Study Can Tell Us. In Proceedings of the 2nd International Conference on Human Aspects of Information Security, Privacy and Trust. Heraklion, Crete, Greece, 195–206. (2014)
27. Lee P. Y., Choong Y. Y.: Human Generated Passwords – The Impacts of Password Requirements and Presentation Styles. In Proceedings of the 3rd International Conference on Human Aspects of Information Security, Privacy and Trust. Los Angeles, CA, USA, 83–94. (2015)
28. Komanduri S. et al.: Of Passwords and People: Measuring the Effect of Password-Composition Policies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Vancouver, BC, Canada, 2595–2604. (2011)
29. Adams A., Sasse M. A.: Users Are Not the Enemy. Communications of the ACM, Vol. 42, No. 12, 40–46. (1999)
30. Milosavljević G., Sladić G., Milosavljević B., Zarić M., Gostojić S., Slivka J.: Context-sensitive Constraints for Access Control of Business Processes. Computer Science and Information Systems, Vol. 15, No. 1, 1–30. (2018)
31. Lorenz B., Kikkas K., Klooster A.: 'The Four Most-Used Passwords Are Love, Sex, Secret, and God': Password Security and Training in Different User Groups. In Proceedings of the 1st International Conference on Human Aspects of Information Security, Privacy and Trust. Las Vegas, NV, USA, 276–283. (2013)
32. Grawemeyer B., Johnson H.: Using and Managing Multiple Passwords: A Week to a View. Interacting with Computers, Vol. 23, No. 3, 256–267. (2011)
33. Zhang Y., Monrose F., Reiter M. K.: The Security of Modern Password Expiration. In Proceedings of the 17th ACM Conference on Computer and Communications Security. Chicago, IL, USA, 176–186. (2010)
34. Shay R. et al.: Encountering Stronger Password Requirements: User Attitudes and Behaviors. In Proceedings of the 6th Symposium on Usable Privacy and Security. Redmond, WA, USA. (2010)

35. von Zezschwitz E., De Luca A., Hussmann H.: Survival of the Shortest: A Retrospective Analysis of Influencing Factors on Password Composition. In: Kotze, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler M. (eds.): Human-Computer Interaction - Interact 2013. Lecture Notes in Computer Science, Vol. 8119. Springer-Verlag, 460–467. (2013)
36. Qiu W., Gong Z., Guo Y., Liu B., Tang X., Yuan Y.: GPU-Based High Performance Password Recovery Technique for Hash Functions. Journal of Information Science and Engineering, Vol. 32, 97–112. (2016)
37. Gosney J. M.: 8x Nvidia GTX 1080 Ti Hashcat Benchmarks. (2018). [Online]. Available: https://gist.github.com/epixoip/ace60d09981be09544fdd35005051505/ (current July 2018). Archived by WebCite® at http://www.webcitation.org/70sbkefP5/
38. EUR-Lex. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. (1995) [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML (current July 2018). Archived by WebCite® at http://www.webcitation.org/70sc3rROv/
39. St. Clair L. et al: Password Exhaustion: Predicting the End of Password Usefulness. In Proceedings of the 2nd International Conference on Information Systems Security. Kolkata, India, 37-55. (2006)
40. Bonneau J., Herley C., Van Oorschot P. C., Stajano F.: The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. 2012 IEEE Symposium on Security and Privacy. San Francisco, CA, USA, 553-567. (2012)
41. Biddle R., Chiasson S., Van Oorschot P. C.: Graphical Passwords: Learning From the First Twelve Years. ACM Computing Surveys, Vol. 44, No. 4, 1-25. (2012)
42. Florencio D., Herley C., Van Oorschot P. C.: Pushing on String: The 'Don't Care' Region of Password Strength. Communications of the ACM, Vol. 59, No. 11, 66-74. (2016)
43. Melicher W. et al.: Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In Proceedings of the 25th USENIX Security Symposium. Austin, TX, USA, 175-191. (2016)

Leon Bošnjak is an assistant and researcher at the Faculty of Electrical Engineering and Computer Science of University of Maribor, Slovenia. In 2014, he received his master's degree in Informatics and Technologies of Communication, and is currently a doctoral student of Computer Science and Information Technologies. His research interests include system security, textual and graphical passwords, and authentication methods.

Boštjan Brumen received doctor’s degree in informatics in 2004. He is an associate professor at University of Maribor, Faculty of Electrical engineering and computer science. He was Secretary General (Provost) of University of Maribor for two consecutive terms between 2004 and 2011. His research interests include intelligent data analysis, automated learning and learning models, data security and data quality. He has published several articles about passwords in prestigious journals, among them Journal of Medical Internet Research.

Received: March 28, 2018; Accepted: August 25, 2018