About Passwords

András Keszthelyi

Óbuda University, Károly Keleti Faculty of Economics, Institute of Organisation and Management, Népszínház u. 8, H-1081 Budapest, Hungary
E-mail: keszthelyi.andras@kgk.uni-obuda.hu

Abstract: In our age of cyber war and cyber crime, it is critically important to select and use "good" passwords to protect user accounts. A well-known general rule says that passwords should contain a mix of letters, numbers, and special characters. In this paper I will show mathematically that this rule is a misbelief. Instead of this, the length is the significantly important attribute. Then I will analyse the most common password structures and give an estimation on the time requirements of brute force attacks. (Un)fortunately there are a lot of password lists originating from a lot of intrusions and data thefts to analyse, and we have the incredible results of the latest brute force experiments. On the basis of these calculations we can state that passwords can give us strong protection if we apply some simple rules, unless the password encoding algorithm of the operating system is too weak. It is worth the time and energy for mathematicians to develop stronger hash functions and OS manufacturers to apply them, but this is not discussed here, and nor is how password using habits have changed.

Keywords: password; user authentication; password cracking; brute force attack; good password

1 Introduction

Latest news: Twitter was cracked at the beginning of February, 2013. In this data breach, the attacker(s) got access to the shadow passwords of about 250,000 users. Bob Lord, director of Information Security at Twitter said, "Make sure you use a strong password – at least 10 (but more is better) characters and a mixture of upper- and lowercase letters, numbers, and symbols... For more information about making your Twitter and other Internet accounts more secure, read our Help Center documentation or the FTC's guide on passwords." [1] At this point we have a really good occasion to think over what we know, sometimes incorrerctly, about the use of passwords and the attributes of a really good password.

It is a fact that nobody can reach a 100% security level in any field of life, especially not in IT. In our digital age, data security is very important (Google returns 13 million hints) in general, and in user authentication in particular.

“Passwords are a very poor authentication method. It is widely estimated that the majority of security breaches – as much as 80 percent – are attributable to persons picking “weak” passwords that are easy to guess or to stolen passwords that are compromised because of poor password protection practices. The method survives because it is still generally cheaper than the alternatives.” [2] As we shall see, passwords are *not* a “very poor” authentication method if used correctly. At least not poorer than a lost or stolen cellphone, token, etc.

The number of password-protected accounts an average user has is bigger than one would think at first glance, and it is increasing. According to Symantec, 44% of users have more than 20 password-protected accounts. [3] “The average user has 6.5 passwords, each of which is shared across 3.9 different sites. Each user has about 25 accounts (...) and types an average of 8 passwords per day.” [36]

As passwords will not disappear in the future, it is worth the time and energy to learn how to use them correctly and efficiently and to understand why some well-known rules and words of advice are misapprehensions, not to say “urban legends”, such as a “good” password looks like W@fK41#a&2s?.

2 Background

2.1 How to Get Somebody's Password

To store passwords in plaintext form is a serious security flaw. If anyone, anyhow can get access to the file plaintext passwords are stored in s/he will be able to personalize any of the regular users with the utmost ease. To avoid that, in most cases and in most systems the hash of the plaintext passwords are stored instead of the plaintext ones. Hash functions are one-way functions, which means that it is easy to calculate the hash of a plaintext password but it is impossible to calculate the plaintext password from the hash. These hashes are called shadow passwords. “Because the stored passwords cannot be deciphered, they are completely safe, even if the entire password file is (accidentally or maliciously) disclosed.” Denning said in 1982. [4] In those times perhaps there were no computational capacities which would have been enough to do brute force attacks against shadow passwords.

There are two different kinds of possibilities to get other users' password, stealing and guessing. *To steal* a password one can use any tools which are adequate to the circumstances. From hidden cameras to hardware keyloggers and trojan horse programs, there are a lot of possibilities, of which social hacking is not the worst: “Ninety per cent of office workers at London's Waterloo Station gave away their computer password for a cheap pen, compared with 65 per cent last year.” [6].

The other possibility is *to guess* the password. The simplest cases are when the password is a “default” one (e.g. password, asdfgh) or is in close connection with the login name (login – login12) or with the user's person (date of birth). A Frenchman even succeeded in breaking into Barack Obama's twitter account in 2010 (and there are other examples as well): “Cousteix managed to break into the accounts by searching information that is most commonly used for passwords, such as birth dates or pet names, on social networking sites. He lives with his parents and has no college degree, and has not had any special computer training.” [7]

“In 2008, the then-unemployed man was using Skype (...) when he dialed a random number and then entered the code "123456" (…) Although he didn't realize what he had done, the man was granted access to the French central bank's debt service.” [10]

2.2 Some of the Latest Bigger Password Thefts

2009, Hotmail. The list of stolen passwords initially contained 10,028 entries. After cleaning up the list, 9,843 valid passwords remained, of which 8,931 (90%) were unique. The most common password was: 123456. [13]

2009, Rockyou. In December, 2009 32 million passwords were revealed by a successful SQL injection attack. The passwords were stored in cleartext in the database, which is a serious case of carelessness. “The data provides a unique glimpse into the way that users select passwords and an opportunity to evaluate the true strength of these as a security mechanism. In the past, password studies have focused mostly on surveys. Never before has there been such a high volume of real-world passwords to examine.” [14]

“About 30% of users chose passwords whose length is equal or below six characters. Moreover, almost 60% of users chose their passwords from a limited set of alpha-numeric characters. Nearly 50% of users used names, slang words, dictionary words or trivial passwords (consecutive digits, adjacent keyboard keys, and so on). The most common password among Rockyou.com account owners is “123456”. [14]

2011, China. At the end of the year, about 100 million plaintext passwords were revealed from different Chinese websites. The passwords were stored as plain text in this case, too. Some interesting results of analysing the data are: “(1) users might choose less secure passwords for their convenience and ease of memorization, though their primary concern is online security; (2) for the same reasons, password reuse is common, as users tend to use the same passwords for multiple online accounts; and (3) passwords usually contain common words, or personal information, such as birthdays and family member names.” [15]

2012, Yahoo. In the summer a list of 450,000 usernames and plaintext passwords were revealed. Once again: plaintext passwords. According to [16], the top 10 passwords were: 123456, password, welcome, ninja, abc123, 123456789, 12345678, sunshine, princess, qwerty. The top 10 base words were: password, welcome, qwerty, monkey, jesus, love, money, freedom, ninja, writer.

2012, Philips. Only 400 real life shadow passwords were stolen. A researcher tried to crack these with an interesting result. He used John the Ripper to crack the passwords. The number of successfully found passwords shot up, then stabilized, and then remained steady. The first 25% of the passwords fell in 3 seconds; first half of them were found in 50 minutes; and only 53% in total after two hours. [37]

2012, LinkedIn. The data of 6,5 million users were stolen.

2.3 A Functional Approach to Password Usage

There are theoretical approaches based on entropy, but now let us prefer the practical point of view. We can state as axioms that a password *must* be not only hard to guess (for intruders) but easy to remember (for users). At first glance one would think that these two requirements are opposite to each other. A survey in 2010 points out the facm that users know this clearly, at least in theory, see Table 1. [38] In practice there may be problems.

Table 1

Which of the following are the most important factors when selecting a new password? Mark all that apply

Easy to remember	46%
Short and easy to enter	8%
Fun or interesting	9%
Strength (i.e. hard to guess)	71%
Other, please specify	7%

What are the possible ways to guess someone else's password? There are four traditional methods to do that guessing.

The case of default passwords; these may be factory default ones (wifi, switch), or those of a lazy system administrator or user: password, asdfgh, 123456, etc. See, e.g. the case of the French bank, mentioned above. [10]

The case of connection between the login name and password: there is a a formal or a logical or a personal connection between login name and password, For example, login – login19, or romeo – juliet, or Obama – president. See for example the intrusion into Obama's Twitter account above. [7]

The dictionary attack: the attacker collects possible, frequently used passwords into a list and a program tries them one by one at a slower speed (online) or at a higher speed (offline, when shadow passwords are stolen somehow). There are quite a lot of real-life passwords out there that can be obtained easily; see the above mentioned examples and others. This method is useful especially when the attacker wants to crack as many accounts as s/he can – among a lot of people there always will be enough who use simple passwords.

The brute force attack: when the attacker applies a program to try *all* the possible character combinations as passwords.

In short, a "good" password is one when none of the mentioned cracking methods would be successful, or at least not in a reasonable time period.

The ease of remembering our passwords is also not a simple problem. According to the publicly known offline cracking speeds (below) we have to remember quite long passwords. There are techniques which can help you to generate passwords which are easy, or at least easier, to memorize. "The third folk belief is that random passwords are better than those based on mnemonic phrases. However, each appeared to be just as strong as the other. So this belief is debunked. The fourth folk belief is that passwords based on mnemonic phrases are harder to remember than naively selected passwords. However, each appeared to be just as easy to remember as the other. So this belief is debunked.“ [20]

3 Discussion

In this part I am going to point out that some of the widely applied rules are false; first of all, that a strong password ought to contain all kind of characters and must be totally meaningless, perhaps a sequence of random characters. Second, HP published a password generation method that gives a false feeling of security; its weakness is apparent not just today, but it must have been considered as a security risk even when it was published ten years ago.

3.1 The Basic Character Set – Examples

The most important rule nearly everywhere is that passwords must contain all kinds of characters: lower and upper case letters, digits and punctuation marks, or other special characters. This, by itself, is simply not true.

First, let us see two theoretical examples from higher education. "A password based on only small letters, capital letters or numbers has a small key-space. This makes it more easy for brute-force, just because it limits the possibilities." [21]

József Ködmön, associate professor at the University of Debrecen, Hungary, whose research field is cryptography and data security, says that the ideally good password does not contain any meaningful words or expressions and should contain different kind of letters, digits and special characters. [22]

Some examples from the practical life follow. First of all, see Bob Lord's post on Twitter blog, cited above, in the introduction. [1]

The Gmail recommendation: “Use a password with a mix of letters, numbers, and symbols. There are only 26^8 possible permutations for an 8-character password that uses just lowercase letters, while there are 94^8 possible permutations for an 8-character password that uses a combination of mixed-case letters, numbers, and symbols. That's over 6 quadrillion more possible variations for a mixed password, which makes it that much harder for anyone to guess or crack.” [23]

Google Password Help says: “Tips for creating a secure password: Include punctuation marks and/or numbers. Mix capital and lowercase letters. Include similar looking substitutions, such as the number zero for the letter 'O' or '$' for the letter 'S'. Create a unique acronym. Include phonetic replacements, such as 'Luv 2 Laf' for 'Love to Laugh'.” [24]

The Federal Trade Commission (referenced by Bob Lord in Twitter blog) says: “Make your password at least 10 to 12 characters long, and use a mix of letters, numbers, and special characters”. [25]

Twitter: “When you set up your account, be sure to choose a strong password (at least 10 characters that include upper and lower case characters, numbers, and symbols).” [26]

Why is this rule not true? If an attacker could manage the crack with one of the first three guessing methods, the structure of the password would not be interesting, or in other words, the user was careless. So let us suppose the case of a brute force attack. In this case we can be sure that the attacker will find the password – if s/he has enough time. This means that we must take into consideration the number of the different character combinations that the attacker must try in order to find the real password and the speed s/he can provide for the attack.

If one increases the number of the elements in the basic character set, the number of all the possible combinations can be calculated by a power function (x^a, where x is the number of the possible characters and a is the length of the password). Instead of this we can increase the length of our password, which means that the number of the combinations will be determined by an exponential function (a^x). It is well known that exponential functions shoot up significantly quicker than power functions. So length is more important than the basic character set.

It seems that people usually do not apply the rule of the mixed character set, and it is not as if it was generally known that the length is the more important parameter. See the rockyou.com password structure analyses below.

3.2 Method Recommended by HP

In 2003, HP published and recommended a password generating method to produce different passwords for different sites based on a single and simple password provided by the user. A small program concatenates the single password and the site name, then calculates the MD5 hash value, converts it into ASCII by base64 and truncates that to 12 characters. “1. The algorithm cannot be inverted to discover the user password even if the site name is known. 2. The algorithm is a standard, meaning any implementation must produce the same output for a given input. 3. It is highly unlikely that two different inputs will produce the same output. ... In this example, the unguessable password is qwerty.” [27] The software utility can be downloaded from HP[1] even today.

At first glance it is a big idea; the user can select one easy-to-remember basic word as the password, while the result is different, long and random-like passwords for each account. It *is* marvelous, as long as nobody knows that the user uses this method. But what if the attacker can get some foreknowledge about the user's password generating method, e.g. s/he catches a glimpse of the generator program on the user's screen? In this case, the attacker's situation becomes very comfortable: it is enough to use a short list of basic (pass)words, because the password generating algorithm is known and pretty simple.

3.3 Initials of Poems or Long Sentences

A frequently recommended method is to select a part of one of your favourite poems and use the initials of the words or those of the lines as a password. Ködmön also advises this, with some additions. [22] Computersigh.com also recommends this method, with the extension of adding some complexity by changing some of the letters to upper case ones and inserting some digits [28] or using similar looking substitutes (zero and o, numeric one and letter l etc.). [24] Using this method to generate your own passwords, you should prefer your own sentences to classical poems. Having so many electronic libraries it would not take a huge amount of energy to create a list of possible passwords from initials of the best-known poems. Another method [28] suggests: Start with a sentence or two about ten words total, e.g. “Long and complex passwords are safest. I keep mine secret.” Using the first letter of every word, turn your sentences into an acronym: “lacpasikms” (10 chars). Add complexity, make only the letters in the first half of the alphabet uppercase: “LACpAsIKMs”. Add length with numbers: “LACpAs56IKMs”. Add more length with punctuation and/or symbols: “?LACpAsIKMs)”. This looks like a 14-character-long random password (~10^{27} combinations). It can be done, but I think that most people would not like a password generating method consisting of so many steps.

[1] http://www.hpl.hp.com/personal/Alan_Karp/site_password/index.html

3.4 Random Words with Mnemonic Technique

According to another piece of advice, you ought to select some, e.g. four, words randomly, concatenate them to a single password string and use some mnemonic technique to memorize it. For example, let the four words be 'correct', 'horse', 'battery' and 'staple'. The password would be 'correcthorsebatterystaple'; then try to imagine as if a horse said to you, “that is a battery staple” and then you answered “that's correct”. [29]

Correcthorsebatterystaple.net runs a password generator (built and maintained by Afterlight Web Development) on this basis. The default settings are: four words, a minimum length of 15 characters, the separator is the hyphen, append a random single digit at the end. Considering that the basic dictionary contains 10,000 (English) words, we can calculate the number of possible password combinations: 10^{17}, the same as about the possible combinations of an 8.7-character-long, random-like password. Using some different separators or changing the initials to uppercase would give us a result of about $4*10^{20}$, and this is the same as the number of the possible password combinations of a 10.5-character-long random string. These results do not seem to be very good, seeing the time we would need to crack such passwords.

3.5 Resources Needed to Crack a Password

The difficulty of guessing simply means: how much time would be needed to crack a password? In such a case, “resource” has a complex meaning: the hardware, software and/or any kind of background knowledge, included but not limited to, possessing the shadow passwords. Every piece of background information about the password(s) to be cracked may help the attacker a lot, perhaps too much.

There exist two very different possibilities to crack passwords: we can speak about online and offline password cracking. Online cracking means that the attacker tries to log in to the system using the username s/he wants to crack and tries different passwords. In this case, if the system is run properly, s/he has the possibility of a very limited number of tries and/or time. Any system that has been properly set up will not let you try an unlimited number of passwords, especially not at high speed. This means that an attacker can try a few dozen or a maximum of a few hundred passwords per second.

Offline cracking means that the attacker could somehow manage to get the shadow password(s) and tries to crack them using his/her own resources, of which the computational speed depends on only the hardware-software configuration s/he has.

Let us see the methods an attacker can use against our personal passwords (or those of our company) and their time consumptions.

3.5.1 Default Passwords

Default password mean not only factory default ones (e.g. in some wifi equipment) but those of the lazy system administrators and users. For example: password, password01, asdfgh, 123456, secret, letmein, etc. Use 'top password' in Google for more examples. Every attacker will try the most common passwords as the first step; see e.g. the case of the intrusion into the French bank mentioned above. [10] Using these generic easy-to-guess passwords is an invitation to being hacked, as if you left the starter key in the door of your car. These need only an infinitesimal amount of time.

3.5.2 Logical or Literal Connection

Another common error users may commit is when they select a password that is in connection with the login name or the person. So, the well known rule is, do not use a password such as your birthdate, the names of your children, your phone number, etc. In general, do not use anything that is in connection with your person, especially if this is a public data element. See the case of Obama's Twitter account mentioned above [7], and many other similar ones.

Also, never use passwords that are simple derivations of your login name, e.g. bob – bob12, bob – bob!bob etc. These simple derivations can easily be generated.

These kind of passwords, or at least a carefully selected subset of them, may be used even in an online attack. Using these kind of passwords is also serious carelessness. These also need only an infinitesimal amount of time, too.

3.5.3 Simple Dictionary Attack

The next step of the attacker is the simple dictionary attack. S/he collects the most probable passwords into a list, called a dictionary, and then applies a software tool to check them one by one. This means that it is highly recommended not to use any dictionary words, or more generally, do not use any words or expressions for which Google would return any results (and also do not use a word you entered into Google previously).

A dictionary attack is always applied before the full brute force attack, simply due to the fact that even the largest dictionaries will contain far fewer words than a brute force attack must try. In the case of a system that is run normally, a traditional dictionary attack cannot be used online, because there must be some time delay between the failed login attempts, and too many failed attempts will trigger a security alert for the system administrator.

An offline dictionary attack needs little time, as a dictionary contains only a very limited number of words compared to the brute force attack.

3.5.4 Brute Force Attack

The brute force attack means that a program will try all the mathematically possible character combinations.

This kind of attack may be performed only in offline mode, i.e. when the attacker has succeeded in getting the shadow password(s). In this case we can be sure that the password will be revealed, and the only question is this: How much time would it need? In other words: How many tries can be performed in a second and how many combinations has to be tried?

In 2009 commercial products were available that claimed the ability to test up to 2,800,000,000 NTLM passwords per second on a standard desktop computer using a high-end graphics processor. [30]

At the Passwords^12 Conference in Oslo at the end of 2012, Jeremi Gosney demonstrated extreme cracking speeds with a Virtual OpenCL (VCL) that was running the HashCat password cracking program across five servers equipped with 25 AMD Radeon GPUs and communicating at 10 Gbps over Infiniband switched fabric. He could provide an unbelievable 348 billion tries/sec (NTLM password hashes), which means that a 14 character long WinXP password, for example, could be cracked just in six minutes (see Table 2).

Table 2

Gosney's cracking speed

algorythm	*tries/sec*
NTLM	348 billion
MD5	180 billion
SHA1	63 billion
LM	20 billion
bcrypt (05)	71,000
sha512crypt	364,000

Gosney's team was at a point where their implementation of HashCat on VCL could be scaled up to supporting even 128 AMD GPUs. [18] [19]

So this means that the last tool, brute force cracking, can work at a cracking speed of between some thousands and 350 billion tries per sec, depending on the resources the attacker has and the hash function the system uses to calculate the shadow passwords.

3.5.5 Advanced Dictionary Attack

Because brute force attacks may need huge amounts of resources, attackers may want to reduce their efforts, of course. On the other hand, users may want more secure passwords, or at least passwords they think are more secure, without using random strings. There are some well-known methods to do that, so attackers knowing these methods may be able to take advantage of them to optimize their cracking efforts. This results in a method somewhere between the traditional dictionary and brute force attacks, and this could be called as an advanced dictionary attack.

An old, well-known trick is to use similar looking substitutions in the original plain text of the selected password, e.g. password – p@ssw0rd. This method is so well known that it has its own name (“leet”) as an alternative alphabet. Because it is so well known we must suppose an attacker would try this. This method was used in empirical research as well, “...for each word from a dictionary file … make common number substitutions, such a 1 for I, 5 for S etc.” [20] So converting a simple password to leet alphabet does not seem to be a good idea.

There may be other password generating habits, and analysing password structures may help attackers a lot. (Un)fortunately, a very large number of passwords has been revealed (of which I referenced some cases above), so would-be attackers have more than enough ammunition to determine the most common password structures. And we, as well, can do some analyses on revealed passwords to see if there are typical password structures or not, and, if yes, what the most common password structures are.

As described above [14], rockyou.com was cracked in 2009 and about 32 million passwords were made public. I downloaded the list of unique passwords [31] to do some structure analyses. The list contained 14,344,391 unique passwords. There is no formal proof, of course, that this list (or any list) is really the original password list. Only the system administrator could have confirmed the originality of the password list, and only if nobody had changed their passwords between the theft and the sysadmin's confirmation. However, it is *said to be* a real password list and it looks something like that. After some cleaning, i.e. removing lines which seemed to be converting errors (too long lines containing html codes), 14,342,415 items remained, of which the length of 1,789 items is longer than 32 characters. It is not impossible to apply such long passwords, especially if one uses a password manager or copy-and-paste.

Some elementary statistical data follows in Table 3, while Figure 1 shows the most common password lengths.

Table 3
rockyou.com password statistics

average password length	8.74
length <8 characters	33.00%
8 <= length <= 12 characters	59.90%
length >12 characters	7.10%
length >=10 characters	31.03%

The minimum password length is 1. The average password length is 8.7, which might have been just enough in 2009 but is surely not enough today. One third of the passwords were not long enough even at that time. On the other hand, another one third of the passwords had quite a good length of at least 10 characters.

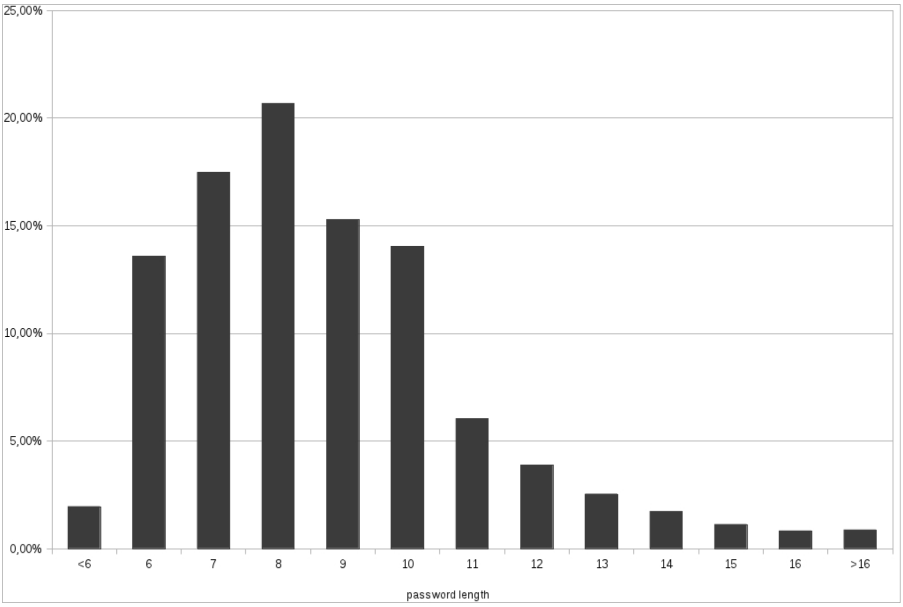

Figure 1
Most common password lengths

I performed some pattern analyses to determine the most common password structures. The character groups I searched for are these: lower case letters, upper case letters, digits, punctuation marks, space, other or special characters.

In the first step, I converted the original passwords, substituting their individual characters according to table 4. So "aaaaa00" means 5 lower case letters and 2 digits at the end, e.g.

Table 4
Character substituting

original	*substituted*
a-z	a
A-Z	A
0-9	0
.,_!?/:;'"-	.
space	_
others	@

Some detailed statistics follow in Table 5. More than one quarter of the passwords consist of lower case letters only. Passwords consisting of only uppercase letters or punctuation marks or others are not preferred; their proportion is less than 2%.

Table 5
rockyou.com password statistics

contains only:	
lowercase letters	26.00%
digits	16.40%
uppercase letters	1.60%
punc. &/or spec.	0.04%
contains space	0.48%
contains at least one:	
uppercase	9.31%
digit	68.08%
punct. or spec.	6.62%
lower+digit	42.36%
lower+upper+digit & none other	2.67%
lower+upper+digit+punc/speci	0.03%

More than one quarter of the passwords consist of only lower case letters. Digits are preferred to uppercase letters or others; two third of the passwords contain digits while only about 16% of them contain uppercase letter(s) or punctuation mark(s) or other special character(s). This means that most people do *not* follow the general – false – rule of passwords, that a password must contain all kind of character types.

Table 6 contains the first 20 of the most common password structure patterns and their lengths. These patterns represent more than the half of the whole set (51.52%). 8 patterns out of the 20 are longer than 8 characters. It seems interesting that these patterns consist of only lower case letters, digits or lower case letters and appended digits. No capital letters, no punctuation marks or special characters.

Table 6
rockyou.com password statistics, most common password structures

structure	*%*	*length*
aaaaaaaa	4.80	8
aaaaaa	4.19	6
aaaaaaa	4.08	7
aaaaaaaaa	3.60	9
0000000	3.40	7
0000000000	3.33	10
00000000	2.99	8
aaaaaa00	2.93	8
aaaaaaaaaa	2.91	10

000000	2.72	6
000000000	2.14	9
aaaaa00	2.04	7
aaaaaaa00	1.91	9
aaaaaaaaaaa	1.87	11
aaaa0000	1.64	8
aaaa00	1.50	6
aaaaaaaa00	1.49	10
aaaaaa0	1.35	7
aaaaaaa0	1.32	8
aaaaaaaaaaaa	1.32	12
total	*51.53*	

Some common passwords of those that consist of only lower case letters are: password (what a surprise!), iloveyou, princess, sunshine, football, superman, zorro, zzzzzzz. Common given names also appear in the list, e.g. michelle, jennifer, etc.

For an attacker it might be a winning strategy to perform a traditional dictionary attack first, applying a word list of 6-12-character-long common lower case words (e.g. monkey, qwerty, nicole, soccer, peanutbutter, sonyericsson, heartbreaker). This would find a lot of the simple passwords among nearly one third (30.06%) of the whole password set. In the second step, s/he would try a word list of six-character-long words with an appended two-digit number (soccer12, summer07, nicole12, etc.).

A brute force attack against shadow passwords ought to be designed like this: in the first step s/he would attack against the 6-10-digit-long numeric passwords. With a cracking speed of one billion tries per second, one would need about ten seconds to finish them (14.59% of the whole password set, nearly every seventh account). Then pattern 'aaaa00' would fall in 0.05 sec and pattern 'aaaa0000' in 4.57 sec (1.5%, 1.64%, respectively).

Then passwords of 'aaaaa00', 'aaaaa00' patterns would need about 32 more seconds (4.97%). At this point, our would-be cracker would need far less than a minute to crack more than one fifth of the accounts (22.69%).

Then pattern 'aaaaaaa00' would follow (a bit more than 13 minutes, 1.91%), then the lower case passwords of 6-9 characters in length ought to follow; and they would capitulate in about 1.5 hours (an additional 16.67% of the accounts).

This means that far less than 2 hours would be enough to crack nearly four-tenth’s (39.37%) of the whole unique password set (more than five and a half million). This seems to be quite effective; there is a chance that the theft of the shadow passwords has not been discovered yet. Longer and/or complicated passwords would need far more time to crack, e.g. the passwords consisting of ten lower case characters would need 1.6 days.

The efficiency of a brute force attack seems to be characterized by a logarithmic-shaped function curve. It shoots up suddenly (with the weakest passwords) then stabilizes (see the Philips case above). As Ducklin says in [37]: “don't be at the left hand side of the graph.”

Supposing that people's password selecting habits change slowly in general, I would say that an attacker could build up quite an efficient cracking strategy by cracking about 40% of the passwords in less than 2 hours.

You can find a much more detailed structure analysis method in [32]. One may be interested in that, especially if s/he wanted to design a very efficient and scalable password cracking software in general.

As for us, we are interested in generating passwords hard to crack, so we need to know the most common structure patterns included in table 6 to avoid them.

3.6 How to Create a Good Password

As “good” passwords must be both easy-to-remember and hard-to-guess, we can formulate some basic rules and methods to help us create “good” passwords. First of all, as you may know best what kind of things you can memorize, use your own method, but keep in mind the possible approaches an attacker might use against your password. Never think that it cannot happen to you. Below some tips follow based on what we have seen above.

3.6.1 Possible Methods

Use a combination of words and some extensions, like computersigh.com's tip above, but use your own method to create your password, a method that is convenient for you and not known to others. Keep in mind that the length of the password is more important than the basic character set it uses, and the structure itself need not be too complicated. For example, as we know, Shakespeare was born in 1564 in Stratford-upon-Avon, so let us combine these together: Shakes-peare1was5born6in4Avon. 29 characters long (30, if the dot is part of the password), not a common structure and easier to remember than computersigh.com's password.

You may also use meaningless words of non-existing languages that sound great at least to you. Drioliano_rodiatenno! – 22 chars.

Prefer foreign, and not English, languages when selecting basic words. Dictionary-based attacks are language dependent. If you have a national keyboard use rare national characters, too, but test it before.

Try some keyboard patterns (but not 'asdfgh' or others like that, of course).

3.6.2 Minimal Password Length

What is a reasonable length for a password in general? You can calculate it easily. Supposing that you have a good enough password which could only be cracked by brute force, calculate the number of possible combinations (the number of characters in the basic character set lifted up onto the power of the length), then divide that number by the supposed cracking speed (measured in tries/sec), then by (60*60*24*365) and you will have the result in years. The bigger the result, the better security you have. Keeping in mind Gosney's results, let the hypothetical cracking speed be 10^{12} tries/sec. Because we can never know... let's multiply it by one thousand and calculate with a value of 10^{15} tries/sec.

For example, see our second example, Drioliano_rodiatenno!. Lower case letters, upper case letters and some others, no digits; the basic character set consists of about 60 characters. As 60^{22}~10^{39}, $10^{39}/10^{15}$/(60*60*24*365)~10^{19}, so there are about 10^{39} possibilities to try, which would need about 10^{19} years. This looks quite good, especially considering the fact that our Universe is supposed to be ~10^{10} years old. And it seems to be not too hard to memorize. A similar password, only 12 characters long, would give us a result of about 70 years. So I suggest that never use shorter passwords and keep in mind: though unbelievable, Gosney's cracking speed is the latest *public* result only.

And do not think it needs too much time to enter such a long password. It needs less time than to unlock your front door, which is a user authentication on the basis the user, i.e. you, has something. You may live in a district where nobody locks their front doors. A computer network is not a world like that.

We suppose at this point that the system for which we use this password is run properly and uses a reasonable hash function to store passwords.

Additional rules of changing your password regularly, etc., must be taken care of, obviously.

3.6.3 Roundabouts, Bypasses

What we discussed above is about how to guess someone else's password or, on the other hand, how to make that too hard. But beware of many kinds of bypasses. You may have the best password in all the world and that will do nothing for you if an attacker, can for example apply a hardware keylogger into the back of your computer (3 secs to install and remove), a small webcam into the ceiling, a trojan horse program into your system, some social hacking, or anything else. See for example the “case study” of the Bastard Operator and the Bank Manager. [32] There is an unimaginably large set of bypasses, so be careful.

3.7 Using and Forcing Good Passwords

In private life it is in our own interest to have strong enough passwords.

In business life it is in our own enterprise's interest. Businesses, first of all, ought to have clear rules, and then make employees keep them. Since any rule is worth only as much as it is kept, businesses should do regular checks. Passwords that the sysadmin can crack with the company’s own resources, which are perhaps very restricted, are really too weak. Online attacks, of course, must be recognized and stopped in time.

We all must keep in mind, especially after Gosney [18] [19] and unlike Denning [4], that if you want security, you have to use physical, administrative and algorithmic protection in parallel. Each chain is only as strong as its weakest link.

3.8 Teaching is Very Important

In our information age it is very important for us to teach not only theoretical material but good practical examples as well. Both the amount of data stored in computers and our dependency on these data increase day by day, so IT security is one of the most important fields in private as well as in business life. As passwords are, and will remain, the first and most important authentication method, we have to concentrate not least on this field.

I, as a teacher, have to draw attention to the fact that the problem shown and discussed above is not simply the problem of only the IT sector, but rather that of education as well. This is not only because passwords are also widely used in education (scholar information systems, e-learning systems, etc.) but because even teenage children may be in danger when they do not know what they are doing when they select their passwords (to log in to gmail, Facebook, etc.). The next generation should also learn the theoretical background. If not, they will not know how passwords work and why it is necessary to select strong passwords or, better to say, what strong passwords are. Students' and children's attention cannot be drawn to such problems too early. If we investigate students' skills and knowledge in the fields of computer sciences here in Hungary or in a wider area, in Central Europe, we find an alarming situation. [34] [35] I, personally, do not think that the situation in the other parts of the world would be significantly better or especially good.

We have to teach our children how to select a reasonably good password and we have at least to try to teach our students that, too. This is not an easy job. A lot of people, unfortunately, prefer laziness to security.

“Results indicate that, in general, users do not vary the complexity of passwords depending on the nature of the site (bank account vs. instant messenger) or change their passwords on any regular basis if it is not required by the site. Users report using lower case letters, numbers or digits, personally meaningful numbers and personally meaningful words when creating passwords, despite the fact that they realize that these methods may not be the most secure." [33]

The China case cited above and many others show that the human factor is the weakest link in the chain of security. This can be developed only by education, by which we can save not only the enterprise property but the privacy of our young children, too.

Conclusions

The most frequent rule of password selection, i.e. that a password must contain lower case and upper case letters, digits and punctuation marks or other special characters, is not true. The length of passwords is a significantly more important factor than the basic character set. Also, any foreknowledge can highly improve the efficiency of a password cracking attack; so it is recommended not to use typical password patterns and/or generating methods. Really good passwords do exist, and can protect your accounts well. Users are advised to learn what hash function their operating system uses to calculate the shadow passwords. If the hash function is weak, that itself will be a security risk, beyond the users' scope. To improve our security level in general, it is important to teach not only the basic rules but their background as well.

References

[1] Lord, Bob: Keeping our users secure, 1 Feb. 2013 http://blog.twitter.com/2013/02/keeping-our-users-secure.html

[2] Cushman, Reid: Primer Authentication of Identity, Project Health Design ELSI Team, University of Miami, 2007, p. 2

[3] Haley, Kevin: Password Survey Results, Symantec Official Blog, 26 Mar. 2010, http://www.symantec.com/connect/blogs/password-survey-results

[4] Denning, Dorothy: Cryptography and Data Security, Addison-Wesley, 1982, p. 162

[6] Leyden, John: Office workers give away passwords for a cheap pen, The Register, http://www.theregister.co.uk/2003/04/18/office_workers_give_away_passwords/, 18 April 2003

[7] Mesquita, Rafael: Frenchman convicted for hacking Obama http://www. boston. com/business/technology/articles/2010/06/25/frenchman_ convic ted_ for_hacking_twitter/, 25 June 2010

[8] Aspan – Baldwin: Sony breach could cost card lenders $300 million, Reuters, http://www.reuters.com/article/2011/04/29/us-sony- creditcards-cost-idUSTRE73S0FL20110429, 28 April 2011

[9] Siegler, Mg.: One Of The 32 Million With A RockYou Account? You May Want To Change All Your Passwords. Like Now., http://techcrunch.com/2009/12/14/rockyou-hacked/, 14 Dec. 2009

[10] Weitzenkorn, Ben: Bank of France's Accidental Hacker Acquitted, http://www.technewsdaily.com/8140-accidental-hacker-bank-france.html, 21 Sep. 2012

[11] Gee – Kim: http://godaigroup.net/publications/doppelganger-domains/, 6 Sep. 2011

[13] Calin, Bogdan: Statistics from 10,000 leaked Hotmail passwords, Acunetix Web Application Security, http://www.acunetix.com/blog/news/statistics-from-10000-leaked-hotmail-passwords/, 6 Oct. 2009

[14] Consumer Password Worst Practices by The Imperva Application Defense Center (ADC), 2010, http://www.imperva.com/docs/wp_consumer_ pass word_worst_practices.pdf

[15] Yang - Hung – Lin: Loose Password Security in Chinese Cyber World Left the Front Door Wide Open to Hackers – An Analytic View, ICEC 12 Proceedings of the 14th Annual International Conference on Electronic Commerce, ACM, 2012. pp. 121-126

[16] Nilsson, Anders: Statistics of "450.000 leaked Yahoo accounts", http://pastebin.com/2D6bHGTa, 13 July 2012

[18] Update: New 25 GPU Monster Devours Passwords In Seconds, http://securityledger.com/new-25-gpu-monster-devours-passwords-in-seconds/, 4 Dec. 2012

[19] New 25-GPU Monster Devours Strong Passwords In Minutes, http://it.slashdot.org/story/12/12/05/0623215/new-25-gpu-monster-devours-strong-passwords-in-minutes, 5 Dec. 2012

[20] Yan - Blackwell - Anderson – Grant: The memorability and security of passwords: empirical results, Security & Privacy, IEEE, Vol. 2, Issue 5, Sept.-Oct. 2004, pp. 25-31

[21] Bakker – Jagt: GPU-based password cracking, University of Amsterdam, System and Network Engineering, 2010, p. 7)

[22] Ködmön, József: Biztonságosabb felhasználóazonosítás az egészségügyben [More secure user authorization in health care], IME - Az egészségügyi vezetők szaklapja [Journal for Managers in Health Care], Vol. 6, Issue 9, Nov. 2007, pp. 46-51

[23] The Gmail Team: Choosing a smart password, http://gmailblog.blogspot.hu/2009/10/choosing-smart-password.html, 7 Oct. 2009, downloaded 5 Feb. 2013

[24] https://accounts.google.com/PasswordHelp, no date of uploading, downloaded 9 Feb 2013

[25] Make Computer Security One of Your New Year's Resolutions, http://www.consumer.ftc.gov/blog/make-computer-security-one-your-new-years-resolutions, 3 Jan. 2013

[26] Keeping Your Account Secure, https://support.twitter.com/articles/76036-keeping-your-account-secure#, 5 Feb. 2013

[27] Karp, Alan H.: Site-Specific Passwords, Hewlett-Packard Company, 2003

[28] Password statistics, http://computersight.com/communicationnetworks/security/password-statistics/, 16 Feb. 2011

[29] http://correcthorsebatterystaple.net/, 12 Feb. 2013

[30] Belenko, Andrei: Password Recovery SolutionsForensics & Investigation IT- Security Audit, Elcomsoft Proactive Software, 16 March 2009

[31] http://www.skullsecurity.org/wiki/index.php/Passwords
downloaded 17 Oct 2012

[32] The Bastard Operator From Hell #7 http://bofh.ntk.net/BOFH/0000/bastard07.php

[33] Shannon, Riley: Password Security: What Users Know and What They Actually Do, Wichita State University - Software Usability Research Lab, February 2006, Vol. 8, Issue 1)

[34] Kiss, G.: Measuring Computer Science Knowledge Level of Hungarian Students specialized in Informatics with Romanian Students attending a Science Course or a Mathematics-Informatics Course / TOJET: The Turkish Online Journal of Education Technology, Volume 11, Issue 4, ISSN: 2146 – 7242, pp. 222-235

[35] Kiss, G.: Comparison of the Programming Knowledge of Slovakian and Hungarian Students / Procedia of Social and Behavioral Science Journal különszám, ISSN: 1877-0428, p. 10

[36] Florencio – Herley: A Large-Scale Study of Web Password Habits, WWW 2007 / Track: Security, Privacy, Reliability, and Ethics, ACM, pp. 657-665

[37] Ducklin, Paul: Cracking passwords from the Philips hack - an important lesson, http://nakedsecurity.sophos.com/2012/08/22/cracking-passwords-from-the-philips-hack/, 22 Aug. 2012

[38] khaley, symantec employee: Living with Passwords, http://www.symantec.com/connect/blogs/living-passwords, 25 March 2010