Version: 1v2

A theoretical view about the frequency of changing the password

The following text should cover some aspects about the question how the frequency of changing the password influences security.

It's assumed that the password is safe. An attacker has no further informations about the password and the password is secure / state-of-the-art (it's long enough and really random).

The attacker uses brute-force-techniques to compromise the password.

1. Extremal values

1.1 Maximum
The secure password is known, when the brute-force-attack performed every possible combination. (It is assumed that the password is highly secure an will be found at the very, very end of the brute-force-attack)

1.2 Minimum
This is only theoretical. If you change your password very,very often without spending time (nearly zero seconds), then it is highly probable that the attacker will get the right password right in the moment he is starting his first attempt.

So the „Endurance“ of the password is between nearly zero (minimum) and the time which is needed to finish the brute-force-attack (maximum).

2. Further aspects

2.1 Unlucky timing for password changing
Your password will be solved soon, because the brute-force-attack has nearly finished testing every possible combination. Although you change your password right now, it will be found by the attacker, because your password is still his last try. The password you used before was the penultimate one.

2.2 Lucky timing for password changing
You change your password. Your new password has been tested by the attacker before. So he has to restart at the beginning.

Is the probability that 2.1 or 2.2 will occure 50% ?

I would deny it!

The longer your password the more passwords of good quality are available. This causes the attacker to spend more time to be successful. A view to the extremals helps understanding this.

A password with only one digit e.g. has only the charcters 0...9. If we determine, that the numbers 0, 1 and 9 are too easy, only 2...8 remain. The rate of good passwords is 70%.

There is no reason, why a longer password should have a different rate then 70% of good passwords, but number of combinations offer more good passwords - in absolute numbers. So it is more difficult for the attacker to find your password.
You see: To select the password which is used, if there are seven good passwords, it's 1:6 that you have a right guess - that's about 17%. If there are 20 passwords it's 1:13 that's about 8%.

The statement, that the rate of good password is 70% or more or less is something a professional mathematician can answer. But it doesn't matter, because it is equal to both cases.

Conclusion: A well placed moment for changing your password raises the securitylevel. This is caused by the fact that probability rises (corresponding with the rising length of the password), that the new password was tested before. So you raise the probability that 2.2 will be yours.

The following arguments also approve to change not so often.

Let's do a retrospect to the example with 0...9. If you change your password more often, after seven changes all good passwords are burned.

If you have a longer password the „pool“ of good passwords is bigger.
Statistically every change burns one of the good passwords. There will be less good passwords remaining. Your protection crumbles.

If there are no further reasons to change your password, it seems that a frequently changing is disadvantageous.

You can try to improove this by using a good password you used before twice. This behaviour is not recommended by experts. Mathematical expectations for this aspect are not processed here.

What means often or seldom?

There is a link on wikipedia (German language):

https://wiki.selfhtml.org/wiki/Sicherheit/sichere_Passwortwahl

where a period of 72 years was mentioned to solve a brute-force-attack over a password with 8 digits by pumping 1.000.000 digits per second. Without doing high-class mathematics it is obvious that a longer password with infrequently changes makes it more likely to give an attacker a hard time.

Keep in mind that a brute-force-attack is not a linear event - it's exponential. This means that it doesn't take 36 years to get 50% of all possible combinations (see above: 72 years are stated to successfully perform a brute-force-attack to an 8-digit password). The true value will be 55-60 years are needed to test 50% of all possible combinations (sorry for not calculating the exact value).
These values are based on the performance available at the time the article was written.

Due to the fact that performance is continously rising, 72 year will be shortend as time goes by.
The point where 50% has been reached is relating. We assume, that the good passwords are uniformly distributed below and above the 50% point. So it will be a good idea to change the password if time reaches the 50% point.
Illustrative: In our example the good password 2...8 are placed around the 50% point - three above and three below 5. I abdicate to use higher mathematics. I see no reason, why this relation should change significant if there are more password.

To convert this to real life: If the password-owner enters school, the next time he has to change the password he will be retired - based on 72 years, especially if the password is of good quality and sesquipedalian. The increase in performance will reduce these periods. Today (year 2017) the 50% point could be at about 40 years.

If you like „The Hitchhiker's Guide to the Galaxy" you probably will select 42 years... ;-)

Frequently changing your password makes it more likely that the attacker will be successful.

The aspect that your login for changing the password implies a risk is neglected here.

So in real life it is not indicated to change your password. But if someone is looking over your shoulder you have been compromised. Other events like your login on an unknown system, where a keylogger is still working or a man-in-the-middel-attack, are things that can't be evaluated here. The testing of millions of keywords will normally be prohibited by the fact that a normal login-system recognizes multiple tries and should react on this by locking or other behaviour. NIST (National Institute of Standards and Technology) recommends to avoid periodical, forced changes.

Thanks for reading!

Notice:
This essay isn't complete or absolute right. It is no scientific paper. At the end of the day it's you, the user, who is responsible for what he is doing. Technical changes or future implementations can cause significant changes to these aspects.