Fred Cohen & Associates - Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Change your passwords how often?

I get more and more requests for discussions relating to an article I wrote in September of 1997 titled “Change Your Password – Do Si Do”.[1] So I have decided to do an update to reflect modern times. Here is a quote from the original article:

> *But I could be wrong – and you could prove it to me. In the beginning, this was a search for a reasonable basis for making audit recommendations regarding password changing frequency – and it still is. Right now, unless there is a special circumstance, changing them even once in a while seems to me like a poor idea.*

And the conclusion this time is... the same as last time. Times and usage styles have changed, and there are some special situations that may marginally benefit from periodic password changes, but there is almost always a better option. But the devil is in the details...

What's the fundamental question here?

The fundamental question that I think must be addressed is:

> If the goal is to improve the effectiveness of password-based protection, is it beneficial to change passwords more often or not, and if so, how often?

If this isn't the question, look elsewhere for answers. Here is the updated analysis of **reasons to change passwords periodically** and analysis of these claimed reasons:

> **In the limit, if we change passwords on each use, someone watching sessions cannot reuse an old password.** That's true, but if we change passwords every other use, watching sessions has a 50% chance of reuse on each try, and one success may allow the attacker to plant a Trojan Horse for unlimited reentry. The "in the limit" case is not particularly helpful here. If a password can be surveilled technically, it's also likely that the session can be taken over by a man-in-the-middle attack, in which case the password really isn't the problem.
>
> **Changing passwords periodically limits the amount of time that an attacker can access an account if they have gotten a password**. True, but not very important for general purpose systems, where gaining access one time is usually enough to plant a Trojan horse and allow reentry and/or ongoing exploitation. Allowing entry for only a few days is usually enough for significant harm. However, for limited function systems, like many Web portals, and for rate limited access, like computational capabilities and some databases, changing passwords will limit the time of use, and thus the harm. In this case, a risk-related calculation may be useful in understanding the implications.

1 F. Cohen, “Change Your Password – Do Si Do”, Network Security Magazine as part of the “Managing Network Security” series, September, 1997. This article is largely a reprise of the cited article. Rather than referring back to that article again and again, we will shamelessly copy without further citation. The reader is advised to review the previous article – online at all.net.

Changing passwords periodically makes password guessing harder. This is not true for non-trivial passwords. It is only true until the number of passwords that can be guessed between changes becomes a significant portion of the total password space. For example, an 8 symbol passwords generated randomly from 100 choices per symbol, there are 10^{16} possible passwords. There are two cases to consider:

> **External guessing:** From an external interface allowing a password guess every second for any given user identity, assuming no response process, 31,536,000 (~$3*10^7$) attempts can be made per year. So in a year, the chances of guessing such a password is about 1 in 300 million, assuming the same password is never guessed twice. The only case where it is any advantage at all to change such a password is when the new password was already guessed, assuming the same password won't be guessed again, a very bad assumption. Assume we change passwords every day. If we happen to use a previously guessed password (< 1 in 300 million chance for the 1st year of guessing), when we change the password again the next day, the chances of the new password being previously guessed is less than 1 in 300 million. There is no advantage to changing passwords against external guessing, even with this bad assumption.[2]
>
> **Known password file (internal) guessing:** With access to an encrypted version of a password file, programs may be run to guess passwords at far greater speed. There are basically three modes of interest. **(Mode 1)** Guessing can be done so quickly that an 8-character password can be guessed with rainbow tables or a similar technique in a matter of seconds. Unless such passwords are changed every few seconds, changing passwords will then have no useful effect. In order to resist such attacks, passwords have to be far longer. So let's suppose that they are far longer. **(Mode 2)** Passwords are made long and complex enough that guessing takes a long time even with an encrypted password file. For example, suppose it takes millions of years. If this is the case, the analysis is the same as for the external guessing case for 8 symbol passwords, and changing passwords in any reasonable time frame is not useful. **(Mode 3)** The password length and difficulty of guessing is such that the threat of interest can get to a reasonable chance (whatever that is) of guessing a password just after the password is changed. If this is the case, the attacker has to reset their search by using the new password file whenever it changes. Then they will, effectively, be able to continue guessing for as long as they like with the same results as the other cases. It could be argued that an attacker might only be able to get the password file once. Then, changing the passwords before the attacker can guess them will stop the attack.[3]

The only place changing passwords against the possibility of password guessing attack makes sense is when the attacker can get the password file once and only once. And when that is the case, changing the passwords should be done quickly, presumably far faster than once a quarter or once a month.

---

2 For a fee we can come up with worse assumptions and eventually find some where periodic password changes may be justified. Of course for another fee, we are available to challenge those bad assumptions.

3 If you are running that close to the edge, improve password quality and make time-to-guess larger.

An insider with special knowledge about a person might be able to guess enough passwords to break into their account if the password weren't changed often. It turns out that all of the factual information that you could reasonably gather to help you break into a person's computer account could be guessed in short order. Let's say we can gather a list of 10,000 facts that could be combined in 100 different ways per fact to generate guesses at passwords. That comes to 1 million guesses – about the same number of guesses required to try all 3-symbol passwords in the example above. So for an internal attack, unless they change passwords more often than every few seconds, all of these guesses can be tried before the password changes. For an external attack, it will still take more than 10 days of trying a new password every second to exhaust the space. The notion that we would allow unlimited guessing of passwords at that rate is, of course, problematic in any case. But if we had no response in place, and if we changed passwords every day, the odds of a successful guess would still be 1 in 10 every day, and on average, the attacker would guess a useful password within 5 or 6 days regardless of whether the password was changed every day or not. In addition, they would be able to do it again and again. The only effect would be to limit the percentage of time the attacker had the ability to use the password to a half a day, on average. Which means that by forcing the user to change such a password every day, the attacker would only have access, on the average, one day in 10, assuming each action taken requires the use of the password.

If people use poor passwords, changing them more often may have a greater impact on the guessing issue. Of course this is true, but it is not so much a matter of changing passwords more often as it is a matter of choosing hard-to-guess passwords in the first place. It turns out that the effect of password quality on the amount of time to guess is very sensitive. Easily guessed passwords tend to be very easy to guess. In many experiments, it has been found that a password is either revealed very quickly by guessing or only revealed through search times probabilistically in line with the likelihood of exhausting the search space. In other words, almost all easily guessed passwords are guessed by automated password guessing programs in the first few minutes. On a typical system, more passwords are found over the first three minutes than over the next thousand hours. So poor passwords are found too soon to make periodic password changing effective, while other passwords are typically not found for time periods far in excess of the typical password changing times.

Changing passwords is like changing cryptographic keys, and we must change cryptographic keys often according to cryptographic experts. While the latter part of that statement is correct (the need to change crypto-keys), the former part is not normally right. The reason we change cryptographic keys fairly often is that the workload to find the key given a substantial volume of cyphertext (the information encrypted by that key) goes down as we use the key for more information. It is assumed that the attacker is watching all transactions. In the case of passwords, if the attacker watches even one transaction, the key is instantly revealed because it is sent in plaintext. Thus the valid reason for such changes in cryptographic keys is not valid for passwords.

So we see that there are special cases where changing passwords may be appropriate.

Benefits of not changing passwords so often

Risk management is not just about risks. It's also about benefits. And it turns out there are benefits to not changing passwords. Here are some of the more widely accepted ones:

It's easier: It's not easy to remember a random symbol sequence of more than 7 symbols. People have memories that have been experimentally shown to be good for 7+/-3 things. For about half the population, that comes to less than 7 things, and for almost everybody, it comes to less than 11. So there is considerable workload for the average person to remember a good password of substantial length. Various tricks can be used to make it easier to remember longer sequences, at the cost of a smaller space of possible passwords per unit length. The more times we make people go through this work, the harder it is on them. The less often they have to change passwords, the easier it is for them.

People write down new hard-to-guess passwords: When people get a new hard-to-remember password, most of them write it down or put it in their computer somewhere. The more often we change passwords, the larger the portion of the time they will be written down somewhere, and the more likely they are to be written down. When they are written down, they are potential targets of attacks other than real-time surveillance or extraction from peoples' minds.

People may have many accounts to change: With all of the Web sites, services, and computers people now interact with, they tend to have a lot of passwords. If changing one password every month or quarter is a minor inconvenience, imagine what happens when you have to change 100 of them. That means remembering more than one new password per day. If they all have to be hard-to-guess, not written down, and meet complex and varying specifications (i.e., 100 different variations on 8-12 symbols, at least one upper case, one lower case, a digit, and one special character), there is essentially no chance that people will be successful. This means one of three things:

Use the same passwords in multiple places: Studies have shown that this is quite common today. In one study, no less than 20% of passwords used in one place were readily identified in another place the study looked. This means that the security of all of those places is no better than the security of the least secure of them, and risk is aggregated. This then defeats most of the benefits of hard-to-guess passwords and changing passwords periodically, because it is a good bet that one of the places the user uses the password will be easily defeated. For example, one widely used Federal government “secure” system for contracting has a help desk able to tell users what their password is, and many Internet-based service providers don't use encryption in login processes.

Write the passwords down: This goes back to the issue of people writing down passwords. In fact, the most widely used browsers and similar mechanisms now retain copies of passwords and other credentials automatically, fill in the forms without the user needing to remember passwords, and keep users logged in for substantial periods of time of non-use. Which is to say, the security of the user's passwords depends on the security of their computers. Most studies indicate that most computers are compromised from time to time, so this is problematic.

More password recovery or reset operations: This will be discussed next.

Reduced recovery or reset operations and risks associated with them: If people don't write passwords down (or have their computers do it for them), and use different hard-to-guess, periodically changing passwords, they end up doing far more password recovery or reset operations. This means:

Wasted time, inconvenience, and additional password exposure: Every time a password recovery or reset is done, it takes at least a minute of human time. That translates into dollars of lost revenue and expenses. If it is a password reset, it produces another password change, which is to say, it makes the password change problem even worse. If it is a password recovery, it produces an additional transmission of the password to the user, providing another opportunity for the password to be found out.

Reset and recovery are less secure and unchanging: Password recovery and reset operations tend to depend on things that do not change often or at all, are relatively easy to guess compared to even simple passwords. If there is a list of 10 questions for a password reset, they will tend to be the sorts of things that are easily guessed and unaltered, so that the user will tend to remember them when they cannot remember their password. For example, mother's maiden name, name of favorite pet, favorite vacation spot, name of favorite sports team, and so forth. How often do these things change? They are, after all, effectively passwords. And many of the answers are from a fairly limited set of likely possibilities? How hard would it be to guess the favorite sports team of someone from Pittsburgh, PA? Likely not that hard. And pure guessing would, in many cases, require far fewer tries than guessing an 8-symbol password

Denial of service attacks: Since the password recovery or reset operation may be invoked by someone who doesn't know the password itself, it may be exploited to cause denial of services. For example, even if an attacker cannot get a password by issuing a reset operation, they can cause the user to be unable to operate until they act on the reset. If this can be repeated, or worse yet automated, or worse still automated for a large number of users, it can be used to do large-scale denial of services.

Increased risks: If you can't protect the password, how can you protect more information associated with the recovery? After all, instead of protecting a relatively small password, the user and the systems controlling access must remember more things. And these things they are to remember are also things that may be asked for password recovery by other organizations. So the answers to the recovery and reset questions are likely increasingly available at more and more locations, each using the same things to “secure” a password.

People may get a false sense of security: If users come to believe that they are better protected because they change passwords more often, they may come to embrace this as doing their part. Is this really the best way to spend our good will and people resources? Given the option, I would certainly select some other "one thing" to have every employee do on a regular basis. Perhaps doing better backups?

Reason may prevail: If people in charge of security refuse to allow things that don't make sense to become part of their culture and mandates, they may be viewed as being on the right side of the issue. Every time security makes seemingly senseless decisions or enforces an officious policy without a sound reason, they lose the good will and support of those they depend on for actual security. By using reason and being reasonable, security overall is more likely to succeed.

There are real costs in time and overhead, increased complexity, increased risk, and human factors associated with creating such rules and enforcing them. Unless the benefit outweighs the costs, it's not a very good investment. As a general rule, changing passwords on a periodic basis does not seem to make sense. And this has not changed for the last 15 years or so.

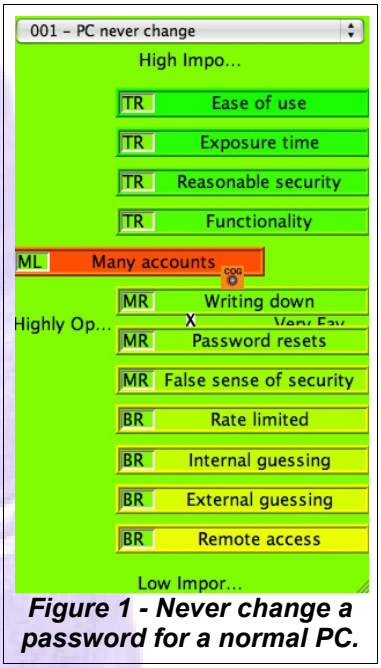

Figure 1 - Never change a password for a normal PC.

Figure 1 shows a typical decision for a normal PC without any special remote access from other locations. While the importance of different factors may vary from system to system, and the level of favorability might change for different situations, the overall decision is not sensitive to any particulars. There simply isn't a good reason we could identify for a user of such a system to have to change passwords with any regular frequency.

But what about the special cases?

Some special cases:

There are several special cases where examination of this issue may require a different approach. Here are some of the ones that pop out.

The password controls a cryptographic communications system: As was briefly discussed earlier, cryptographic keys must be changed periodically if things they encrypt can be intercepted by hostile forces and if time constraints on attacks against keying material are appropriate. This is not the place to go into full detail, but if you want to know more, look into cryptographic key management protocols.

You suspect someone broke in: If you think someone is accessing a system illegally, it might be prudent to change passwords for all users and all accounts for which their passwords may have consequences. After the attack itself has been stopped and the attacker is no longer in control or able to access the system, additional steps are prudent to assure that system corruption hasn't resulted in reentry paths for the attacker, secure evidence for possible legal actions, and so on. After the system is properly secured (or reconstituted), users should change relevant passwords against the potential that the attacker also took and wishes to exploit password information.

A password is shared: While password sharing is generally to be discouraged, there are times when it happens. In these cases, it is important to change the shared passwords every time anyone is removed from the shared access. Similarly, if enough people share an account, changing passwords periodically may be a wise step in providing assurance that users that no longer need to have access do not have such access. Since such systems

normally operate by giving the new password to people only when they ask, those who are no longer using these systems end up without the password after a time.

The enemy has a known capability and you have a known limitation: If you do an analysis and find that an enemy can break some protective barrier through password guessing in a given amount of time, and if you cannot increase this time by normal internal actions, you might decide to change passwords frequently. (e.g., A firewall has a maximum of 6 characters in a password and they are all forced to be upper case letters or digits leading to only 26^6 possible passwords and 1,000 guesses per second are possible and no other protection can be put in place leading to only 4 days to try all passwords. You might decide to change passwords every 8 hours so as to limit the length of access in a takeover.)

Passwords are stored and used online: In cases where passwords to systems are stored online for automated remote access and the operating environments are not well-protected, it may be prudent to change passwords to critical accounts so as to force users to manually type in passwords or to prevent break-ins to machines from granting access to accounts on other machines. This is common in Internet access today and is particularly helpful if it is expected to take substantial time for attackers to locate and exploit passwords after entry.

Some other reason to make the change one-time: It might be valuable to change passwords of critical systems in conjunction with the movement of backup information to other sites. For example, when disposing of a machine that is not properly cleaned before disposal, or when backups are moved to off-site locations after a substantial delay, this has the effect of protecting against exploitation of stored passwords residing on released media.

Limited function remote access with substantial value exploitable over long time frames: For cases, like Web services, things may be quite different, for several reasons. Figure 2 shows an analysis of a situation in which a remote access Web service is offered; within which limited functionality access is provided, there is limited damage linearly increasing with exposure time for those with unauthorized access, and where the value is substantial.

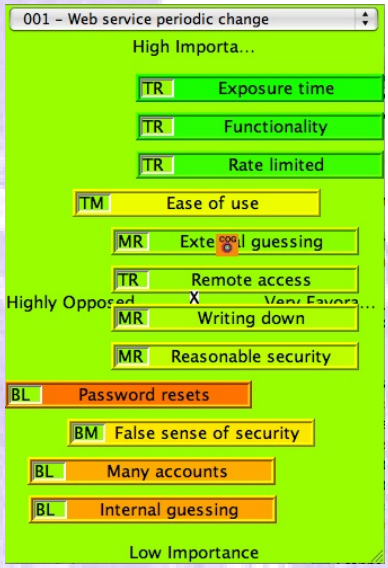

Figure 2 – Periodic password changes for a Web service.

An example review of a special case

In this last special case, an analysis goes something like this:

> **Exposure time:** Exposure tends to lead to linearly increasing harm over long time frames. Thus reducing exposure time to a management specified threshold is beneficial, and faster change times does this against unauthorized password use in many scenarios.
>
> **Functionality:** Limited function implies that password access only grants normal use over the time of exposure. Thus there is no general purpose system issue with planted Trojan horses or reentry after the password is changed.

Rate limited: Rate limiting means that consequences increase linearly with time over long periods. Thus, assuming this is in fact the case, reducing exposure time has benefits at any time it is done.

Ease of use: Web interfaces track passwords for ease of use even when passwords change frequently. So the user experience is not significantly harmed as long as the change rate isn't too high.

External guessing: External guessing is present and often attempted. This will not make such guessing harder, except in that in this mode longer and stronger passwords may be available, but it will limit resulting exposure.

Remote access: Remote access via password is a key issue in limiting the linear loss of value.

Writing down: The user's browser will normally write the password down, but people are less likely to. Access to the user computer will likely grant access regardless, so little is lost by the user's computer writing the password down. And after/if access to their computer is lost, the next password change will reduce consequences.

Reasonable security: There is a sense by the users that you are being reasonable in the security space, especially if the reasons and rational for this rate of change is explained.

Password resets: Few password resets should be necessary with Web-based access, because browsers will remember passwords for the user. But they are still a negative when and if they have to happen.

False sense of security: Changes do have real value and this can be explained. But there is the potential for a false sense of security, so the process should be carefully explained.

Many accounts: People are less likely to use the same password in other accounts because they don't have to remember it, and because it changes frequently when they don't have to change the passwords for other accounts. But this may still happen, and to the extent it dies, it introduces increased exposure.

Internal guessing: Internal attack requires access, and with that access, no defenses will likely remain against Trojan rootkits. But since the passwords change frequently, at least the exposure from external exploit will be limited until the next time the attacker gains access to a password file.

However, before embracing this particular decision, it might be a good idea to understand where the protection comes from. The password change rate is only effective because of the interaction with the rate limiting of the service and the management specified threshold. And even then, only if the specifics are such that the change rate is in an acceptable range. For example, if the cost per unauthorized access is $100, and the rate of loss is limited to $10,000 per day per, they the protection comes from the fact that legitimate users are not permitted to perform more than 100 transactions per day. If the password change rate is once per month, then the loss is limited to $3M (30 days @ $10,000/day) from a single stolen password. And the price is that all of the other users also have to change their passwords once a month.

As we move the change rate to limit losses to $10,000 per successful password exploitation, all users will have to change their password every day. The process requirements for such changes start to become problematic as well, since whatever process is used, that has to be protected from attack as well. If there are a large number of users, then the rate at which the users have passwords compromised also has to be taken into account. For example, for 1,000 users, the worst case loss would be $10M/day even with password change rates of once per day. If only 10 successful password attacks per year occur, then that's $100,000 in worst case expected loss per year, and the loss scales linearly with the number of days between password changes.

But we have ignored the costs of changing these passwords daily. And there are costs. Suppose it takes 1 minute per user per day to change passwords, and the total cost of that change is $1 per user per day. At 250 days/year and 1,000 users, that is $250,000/year in lost time. And at 100 transactions per day, that's about 5 minutes per transaction, and the 1 minute per day of password change time then costs one transaction per user per week in lost transactions, or $5200/year/user in lost transactions. That's $5.2M/year in lost transactions for the 1,000 users. If each lost transaction costs $50, that's $2.6M in losses from password changes. But of course we can simply add more employees to compensate for this, with the resultant relatively small increase in risk from stolen passwords. We won't try to optimize the situation for this particular case, but you can imagine that at one password change every 2.5 days, the costs of password changes goes down and the worst case expected loss goes up. For different specifics, different results will be produced, and calculation is necessary in order to make sense of this. Since the calculation is also subject to estimation errors, a sensitivity analysis is called for.

But even if we carried this through to optimization, that is not the end of the analysis. Just because periodic (i.e., daily) password changes are one approach to limiting loss in special cases like this one, that doesn't make it a good idea for most cases. There are many other special cases where things don't work out as well because the numbers aren't as favorable, or because of other factors in the environment.

Alternative strategies for special cases

In addition to doing this sort of analysis for special cases, there is the challenge of comparing this approach to alternatives. For example, here are two alternative approaches:

> **Daily reconciliation by the user:** Suppose that instead of requiring password changes every day, we required reconciliation of accounts every day. In that case, at the end of each day, each user would get a summary of activities and be responsible for reconciling it to their actual activities. Let's suppose further that the users are our customers, that failure to identify an erroneous activity at the end of a day makes them liable for the results, and that until reconciliation, no actual transactions are completed. Now the situation changes dramatically. There is no benefit to the password changes for the provider, because they have no risk from such an attack. The risk lies entirely on the user, and they have to reconcile every day to protect themselves. If they detect fraudulent activity, they can prevent reconciliation, and if they want, they can change their password to stop it the next day, assuming this was the cause. This is a risk transfer approach.

> **Security token:** A security token that, for example, changes passwords every minute, offers an alternative. If the token is well implemented, password-based frauds of the sort described are completely eliminated. A security token costs less than $1/day/user, including the cost of replacement of lost tokens, etc. In this case, $250,000/year is the total estimated cost. The per transaction time loss is on the order of 12 seconds, for a loss of one minute per week, or about 10 transactions per year per user. That's 10,000 lost transactions per year for $1M/year in lost transactions, or $500,000 in lost net/year for the given scenario. Assuming a lost token can be replaced in 5 minutes (i.e., the workers work in a facility that support the effort) and one lost token per 10 employees per year, this adds 500 minutes per year, or 100 transactions per year in lost time from token replacements (as opposed to the unidentified lost time in password resets from the periodic password change example). At $50 net/transaction that's $5,000/year in lost net from the lost transactions from token replacement time. A lost token that grants actual access will be reported very soon in this case, because the user who normally uses it will be unable to carry out a transaction, and thus the loss from a stolen token being exploited is likely only a few transactions, the offender may be at increased chance of getting caught, and of course transactions can be held for an hour to eliminate the vast majority of such losses. In this scenario, a security token looks like a better choice than periodic password changes, even if the token system is “cracked” every few years.

For this example special case, the alternatives seem to be better choices than periodic password changes. But of course it all comes down to the details.

Conclusions

For general purpose computers, nothing has really changed in the last 15 years or so in terms of the rationality of periodic password changes. But increasingly, special purpose systems are popping up, at least from the standpoint of the users. These are typified by Web services.

There are some increasingly common situations which may benefit from periodic password changes. But finding those situations is non-trivial, they tend to be applicable to very specific circumstances, and they are not sound for general use across a wide spectrum of situations.[4]

In special cases where periodic password changes may be useful, there are typically better alternatives. These include various risk transfer (e.g., transferring the risk to the user), reduction (e.g., rate limiting), avoidance (e.g., the token approach), and acceptance (e.g., the residual risk is always accepted) approaches. If left unexplored, you risk offending your user base in the name of “security”, and losing the opportunity to gain their assistance in other areas and acceptance of more effective or necessary protection procedures.

While the specific “back of the envelope” calculations provided here are far from complete, a sense of the factors in the decision and the situations in which they favor one or the other decision may be gleaned. We hope they will be useful. Perhaps more importantly, the decision-making process will hopefully become something that involves thought and analysis, rather than a rote approach to a discipline that is unpopular and rarely worthwhile.

4 In one case that was identified, a password from an important application was also used for an unimportant test application, the test application didn't encrypt passwords, the unencrypted password file was put on a public server in close proximity to the important system, and was left there for 9 months before exploited.

Counterpoints, rebuttals, and surrebuttals

Matthew Rosenquist of Intel had some counterpoints we thought readers would be interested in. Here are his counterpoints, our rebuttals, and his surrebuttals (where offered).

Counterpoint 1. Forcing employees to reset their password has behavioral upsides. It improves security awareness, bestows a positive feeling of empowerment (also a negative feeling of frustration at times, but that is another discussion), and reminds them they are held accountable to responsibilities outlined in the corporate security policy: (i.e., their involvement is crucial to securing the enterprise). It reinforces the teamwork element all employees contribute to collectively secure the organization.

> ***Rebuttal:*** *By this argument, any useless security practice that forces users to do something is a benefit. How about if they repeat the corporate policy each day in order to get in the front door? Why not use something that is effective as a security practice as the way to get employees to be more aware and feel more empowered?*

Counterpoint 2. From the perspective of disrupting attacks, it impacts the path of least resistance. Knowing an organization has strong password controls, both behavioral as well as technical, may deter or redirect attackers to different methods. And anytime I can keep bad people from targeting my domain controllers or password files, that is a good thing. Even if it is only some of the time.

> ***Rebuttal:*** *The notion that telling attackers that you have this policy will deter them is problematic on several fronts. You could simply claim it without making the users do it, and that would have the same deterrent effect. And of course, how do you know you aren't convincing attackers to do something that will be more likely to work instead of less likely? After all, password guessing is a pretty poor way to attack systems compared to other methods often used.*
>
> > **Surrebuttal:** False claims are dangerous and work only until the truth is revealed. Then the reputation of your security efforts are undermined. It also is worthless for the Internal threats who would realize strong password controls are a façade.

Counterpoint 3. The value of compromised passwords degrades over time with a strong reset password policy. So if someone wants to sell such data on the black-market, it would play havoc with their profit margins, which again may deter buyers and even those wanting to sell. It is akin to selling a product (ex. gallon of milk) with a very short expiration date. Customers do care and will seek products with longer shelf life.

> ***Rebuttal:*** *I find no evidence that the actual value of goods has anything to do with their price on the black market or elsewhere. It is perceived value. But perhaps more importantly, you are making your passwords more valuable if they are time limited. You re saying to the buyer "this is hot stuff – buy now while it's still good – premium price" and to the seller "break in again and again – install a more permanent attack method to grant access". If you are claiming protection by perception you can just as well claim the opposite – increased motive for more vigorous attack because of the increased perceived value.*

Counterpoint 4. You indicated not much has changed in the past 15 years... Moore's Law of computing power for brute force attacks has been in effect and therefore the time to compromise passwords has changed greatly. This is paramount and should not be overlooked. Today, someone can rent supercomputer processing power on the cheap, to crack a password file. (As an example, Amazon's EC2 has been used for this).

> ***Rebuttal:*** *For external attack, Moore's law changes nothing. For internal attack it doesn't matter much. The value of changing a password only comes if the guessing time is between the margin of too short (you have to change them too often) and too long (you don't have to change them very often at all. If too short is a minute and too long is 10 years, this is only a factor of 5.26M, or about 22 bits (less than 3 bytes). So by adding a few symbols to the minimum password length, you eliminate any utility of changing it.*

Counterpoint 5. Brute force attack against a single users, to get a few of their passwords is not too worrying in general. But doing such against a corporate password file of 100k users, where such access could expose billions of dollars in loss..... Yeah, your numbers don't account for such situations.

> ***Rebuttal:*** *Right you are. There are special cases where more protection is needed. But in this case, is a password that you have to change every day good enough? If you put that much risk on a single password, or any other single form of authentication, you are making poor risk management decision. Changing passwords more often is not the solution here.*

Counterpoint 6. Believe it or not, keeping people involved in manipulating their own password improves the response to incidents. First, because of the comfort factor it gives users another mechanism to respond when they may (emphasis on 'may') believe their accounts have been compromised. Second, if instructed, users familiar with the process can efficiently change their own password with fewer calls to the technical assistance team.

> ***Rebuttal:*** *So does this mean it is better to have them change passwords every 15 minutes? Every 5 minutes? Every second? How often is often enough? And is changing passwords really the approach to this you would prefer? This only practices them in one thing that is rarely the thing they actually need to do. And do you really want users to change passwords whenever they think their accounts may have ben compromised? Won't that create havoc when an email rumor spreads? While I understand that practice makes perfect, practice costs, and imperfect may cost less.*

Counterpoint 7. We have seen password files hacked in the industry. I won't go into details, but an important step to control the risks is to then force a password reset by users. If a mechanism is already in use and familiar with users, it is much lower of an impact.

> ***Rebuttal:*** *Again, the question is not whether passwords should ever be changed, but whether they should be changed at a standard rate, and if so, what rate should that be. If you want to justify a particular rate for password changes, what other rates are you specifying for what other security actions, and how much do they cost? Should users have to do a shutdown and emergency patch every week because it might happen some day? Should they have to do a complete system recovery every month? It is like*

fire alarms, we do it every other Thursday because it keeps us in practice? Should we practice calling 911 every week because there might be a heart attack? If you are going to justify password changes on this basis, you should be prepared to do the comparable thing for every other security and safety function and justify the relative rates based on the relative risks.

Counterpoint 8. Weakest link position applies to passwords. If compromised, it opens the door to an attacker to begin escalation of access activities. Constantly keeping the passwords fluid within the organization can disrupt this activity.

Rebuttal: *Password changes at rates don't have any effect on this. If an attacker can escalate the attack, they can do it on break-in the first time. Attackers readily plant reentry capabilities within seconds. There are free toolkits to do this once entry has been accomplished. And password changes after that are of no real value.*

Counterpoint 9. If the accounts are compromised but not detected, a password reset can throw a wrench in the works for the attacker. In fact it can be a Detection mechanism (Defense In Depth reference). Such situations may lead to a lockout of the account, which may flag someone to look into the matter. This has happened more than once. Was talking to a bank last week about that very situation which ultimately detected unauthorized account access.

Rebuttal: *Again, there are select instances when this has some benefit, but it has to be weighed against the costs and the extent to which this "detection" mechanism is effective. You seem to be saying that the attacker had unlimited access for a month or two and was then stopped because they failed to escalate their attack in that time. But instead of requiring password changes every month for everyone, suppose I had better detection of misuse? Would I catch more or fewer attackers? Maybe I would have detected this attacker sooner.*

Counterpoint 10. Password resets protect against some risks of over-the shoulder password surfing. My kids try to do this to me the little buggers, with my locked down personal iPad (I limit their access to it). They catch glimpses and over time mentally piece together the passcode. Eventually, they will get the whole thing as they go through the process of focusing on different sections (beginning, middle, end) as is human nature. With regular password changes, in addition to other controls, time will work against them. Buy by the time they get the last section, the first would have changed. Let's look at a more innocent situation. We sometimes are near co-workers when they login. It is normal to accidentally see them type in part of their password. We may catch the first part, what it ends with, maybe notice the numbers are in middle, etc. Over time, we could mentally piece it all together, even without every trying combinations on their system. Regular password changes disrupt this process.

Rebuttal: *Again, this is a poor approach compared to the alternatives. Suppose that I simply warn the user of failed attempts to login to their account. Then, after the first attempt, the user would be made aware of the attempt. And after each subsequent try they will be made aware again. If they are well trained, they will report this (and the computer will automatically report it anyway), and the perpetrator will be caught and punished.*

Summary rebuttal

There is a far more important point to be made here. The notion appears to be that we should use this method of inconveniencing all of the users because sometimes it helps with some things that are largely personnel performance and behavioral problems with other users. That seems to be a major problem with security overall, and one that should be addressed. If some small number of people do something undesired, we inconvenience everyone else to allow the misbehaving individuals to continue to misbehave. Instead, we should perhaps focus on stopping the misbehaviors.

> ***Summary surrebuttal:*** *Changing passwords every 60 or 90 days, which seems like a typical industry number for professional organizations, can be viewed as an annoyance. But spending less than a minute changing a password every few months seems like a small price in comparison to the benefits.*

The question of continuous equations for a decision

Several readers identified that it would be far more useful to have equations to help in the decision process about how often to change passwords for different situations. There are a few issues here.

Not a continuous metric space: This article looks at least at ease of use, exposure time, reasonableness, functionality, number of other accounts, writing down, password resets, false sense of security, rate limiting, internal vs. external guessing (i.e., threat model), and access mode (local or remote). Before even thinking of an equation, we have to look at the different variable types, which in this case include nominal, interval, ordinal, and ratio. Which is to say, they are not comparable with a continuous equation.

How about dividing the space into areas where metrics apply? This is effectively done by the description above for a few cases, but certainly not all. But the other cases appear to be trivial and problematic for requiring password changes. For example:

> **Sniffing threats:** In this case there can be no advantage to password changes. In fact, requiring such changes effectively guarantees that all passwords can be observed during the periodic change times.
>
> **Social engineering:** In this case, the threat will or will not succeed at some rate and be detected at some other rate. In high volume within an organization, the chances of being caught are higher as the volume increases. But all of this is roughly equivalent to the modes of external guessing. So depending on the numbers, the same special cases apply. For general purpose systems any entry implies likely ongoing use regardless of password change, while for special purpose systems such as those discussed above, depending on value as discussed above, changes may be a feasible, even if not the best solution.

Quantitative data is needed to assess risks with sufficient precision: While this may be true for the special cases where periodic changes are feasible, for most of the cases, it isn't. The only real case where this seems to make a difference is Mode 2 of external guessing, and there the real value lies in identifying a password length change that would move from mode 2 to mode 3.

Other interesting comments from the community

Here are some other comments, slightly modified, from Larry Wagoner.

> **Poor password formation by humans:** I think one point you are missing is the unrealistic formation of passwords from the human mind when humans are forced to use upper and lower case, numbers and special characters. Let me explain-- there was a study done years ago and the most common password in the D.C. area was "redskins" -- I would suspect in any major city, the pattern would be the same. For instance, the most common password in Dallas would be "cowboys" and Denver, "broncos" or maybe "nuggets". Suppose we enforced a rule that passwords in the D.C. area would have to contain upper and lower case, numbers, and special characters. Wanna bet what the top password would be?
>
> The likely guess would be "R3dsk!ns". In Pittsburgh, it would be "St33!3rs" or maybe "St33!ers" or even "St33!3r5" for those who are really slick -- wow, three tries might be needed. Common substitutions of "3" for "e", "!" for "l", and "0" for "o" and the capitalization of the first character are the immediate knee-jerk reaction to such a policy.
>
> **Escalating delays for failed password attempts:** For external guessing of passwords, as you know, most systems worth anything will lock an account either permanently or for a length of time (and some lock out for increasing lengths of time -- first lockout is 1 minute, second is 2 minutes, third is 4 minutes, etc.). So even though you state "with no response process" to ignore this scenario, I think it really dilutes your point. So inserting a sentence instead of the phrase "with no response process" to state that this is unrealistic in the real world and that it's just to simplify your math.
>
> **Multi-factor authentication with "weak" passwords:** You may want to give a mention to two factor authenticators -- a hardware token and a relatively easy to remember password -- that is, don't need a 20 character unbreakable password, when a 7 character one might do when combined with a hardware token. Or use of cell phone/text message/smart phone to supply a onetime use password. Anything to transition away from these automagical passwords/secret phrases that are supposed to keep the bad guys away.

And one last comment...

> **Delayed monetization:** One missing analysis point about password theft and "monetization" is that often when passwords are stolen, they aren't used right away. In fact, except for phishing sites that actively man-in-the-middle connections (most common in the presence of second-factor authentication schemes) we find a sometimes substantial time lag between credential compromise, and use. While not in and of itself a justification for changing passwords, it is one data point to consider.
>
> > **Limited analysis:** *This is a worthy point to explore. To the extent that there is a substantial delay between theft and use, if that delay is longer than the password change time, it benefits the defender to change passwords. However, it is also likely that such a tactic will change the markets so that the time to market will be reduced and the value of "fresh" passwords increased.*

