



## Change My Password Again?

*About the insanity of password aging*

Jürgen Pabel, CISSP

It seems that everyone dealing with IT has accepted the necessity that passwords need to be changed on a regular basis. Why is it, that passwords should be changed, and some IT systems even enforce password aging? It appears that no one questions this: password aging is often seen as a technical requirement for password security. However, password aging is only a useful mechanism, if a few specific characteristics apply.

A password needs to be changed whenever unauthorized parties have gained knowledge of it. A password can be compromised in three ways: before it is entered, during the authentication, and through the database that stores the credentials for verification. Commonly, users share passwords because it is the only (or a significantly easier) method for transferring privileges to someone else. For these scenarios the password should be changed right after the the task is completed, but honestly: how many people do you know that actually do that? Of course, it could be that the password wasn't shared intentionally, but rather that it was stolen: users commonly write down passwords in their working area, but even watching someone's keyboard while they enter their credentials is an effective way to obtain access to systems. Even though stealing someone's login credentials by looking over their shoulder could be considered a case of the second type of password compromise (during authentication), this is not the author's intended interpretation. Rather, it's attackers intercepting the credentials after they have been entered into the system. Virtually all IT environments are vulnerable to the most elementary attacks: the threat of keystroke logging devices<sup>1</sup> is virtually left unaddressed in the real world. Further, technically insecure authentication channels are still commonplace and authentication components are prime targets for attackers. If a password's confidentiality has been compromised through one of those avenues, changing passwords is worthless – the attacking party will regain them. Only after such an intrusion has been completely analyzed, the environment's integrity has been restored and the vulnerability that provided the attacker's entrance has been closed, will changing passwords be of any value. However, these recovery steps need to be coordinated in order to prevent the attacker from using any stolen credentials to re-enter the systems – a complicated

<sup>1</sup> KeyKatcher (we are not endorsing this product, but it appears to be the market leading brand)  
<http://www.keykatcher.com>



undertaking.

Authentication attempts need to be verified against a database that stores the user's credentials in order to decide on the validity of the authentication request. Attacking this database is for attackers often the most promising way through which passwords can be compromised, because it potentially yields broad numbers of credentials. Many different technologies are in broad use today to facilitate user data storage and authentication verification: UNIX and Windows can both either use a system-local database, standardized network authentication mechanisms or any custom mechanism. Organizations commonly rely on networked mechanisms to maintain a central storage of all user data in order to achieve data consistency and instantaneous configuration changes. Essentially all modern mechanisms refrain from storing the user's passwords in clear-text, they rather store a mathematically transformed representation of the password. The authentication system applies the same transformation to the provided password during the authentication request and compares the result to the stored value in order to decide upon the authentication. The security attributes of these transformations vary largely, depending on the chosen transformation algorithm and correctness of their implementation. It thus may, or may not, be plausible for attackers to recompute passwords once the database has been retrieved. However, attackers can always attempt to run a dictionary attack on a stolen credential database: they attempt to obtain user passwords by running lists of passwords through the transformation and comparing the results to the stored data in the database – just like the server would, only without relying on the server.

Four threat classifications can therefore be established: user contributions to password compromise, technically intercepted passwords, compromised authentication systems/databases and the low-tech approach of attempting to authenticate using common passwords like “password”, “123”, and the like. The last threat category is commonly countered with password complexity requirements and through forced delays after incorrect authentication attempts. This configuration is common practice, therefore this threat should pose no real practical issue and will thus not be considered any further. Returning to the initial proposition: changing passwords does not necessarily increase security. It has already been illustrated, why this has no effect on compromised password databases, the same arguments hold true for passwords that have been technically intercepted. This leaves user contributions to password compromise as the only classification for which password aging actually improves the situation. IT security departments believe to have understood this issue and use password aging to minimize the time windows during which these authentication secrets have been deprived of their secrecy. However, the only true solution for this problem must be to provide users with easy(!) means for delegating privileges. Most



IT systems either lack these mechanisms or they are too involved for common IT users, therefore enticing users to sacrifice passwords in order to get the job done: time after time, after time. All in all, password aging is at best requiring that malicious parties re-obtain the credentials – it does not enhance the security in any real way, it only causes a bit of overhead, for both the “good” and the “evil”, but the overhead is commonly much higher for the “good” side: IT users and staff. Another aspect must also be considered: users must commonly handle multiple login credentials, and thus are tempted to reuse a single password for all authentication systems. Password aging may also be seen as a solution to prevent this behavior, but: the common IT user wants to remember as few passwords as possible and tends to re-synchronize all his passwords. Without establishing a working environment that honestly values password security, users are bound to slack on their password ethics – and you don't want your user's to give up their credentials for a candy bar, do you?<sup>2</sup> However, fostering such an environment is a meticulous and never ending task, but it may just be worth the effort.

Aside from the user centric password focus, there's a scenario for which password aging makes sense: group accounts. With group accounts, a single user account is used by all group members to log in. Every time a member leaves such a group, the password would need to be changed in order to effectively protect the account from unauthorized access by the departing member. Such group accounts are often used when many dozens or hundreds of users need temporary or restricted access to a resource – enforcing a new password every time anyone leaves that entrusted circle would account for a very high “password churn rate”. This makes immediate password changes impractical; if one only considers the support volume for user's regular passwords, the incurring support overhead for such group accounts would be imposing. Password aging might be a workable solution for managing group accounts; however, the author has never seen password aging being employed for such accounts.

The concept of password aging only works if passwords was obtained through a single occurrence. For all reproducible attacks, password aging does nothing – and the only scenario for which it might present a usable solution that doesn't incur a massive organizational overhead, is not commonly used. Instead of deploying resources to password aging and its effects, invest in educating for your IT users and staff about IT security, as well as better suited mechanisms for authentication and temporarily delegating privileges.

---

<sup>2</sup> Please refer to the InfoSec surveys about users' willingness to share their passwords  
<http://www.infosec.co.uk>