

Experiences with Password Policies

by: Per Thorsheim , 06/30/2005

<http://www.securitydocs.com/library/3410>

Introduction

This article has been written based on my own experiences while performing penetration testing and security audits for large and small organizations domestic and abroad. This article is targeted at providing a better understanding of the weaknesses that often surrounds the choice and change of passwords on both administrative and technical levels, and gives ideas on how to reduce such risks.

A general observation often made is the inconsistency between the documented and the technically implemented password policy. Another general observation is that upon the (batch) creation of several new accounts, weak passwords are used as well as reusing the same password for all the accounts. Last but not least the occurrence of accounts with names such as 'testing' and 'training', which exist in most user databases, and the password rarely is anything else than the username, perhaps with '1234' added at the end, and can be found in seconds using automated techniques.

At a large corporation I was told that an internal questionnaire showed that approximately 75% of all employees said their passwords were in compliance with the password policy. Technical controls showed that less than 2% of them actually were in compliance with the written password policy.

Written vs technically implemented policy

I have over and over again noticed and pointed out the differences between the corporate documented password policy and what's really implemented in their various systems. Sadly these differences almost always end up with the implemented policy being the weakest part.

This problem has two primary causes. Number one is that the written password policy hasn't been communicated well enough to the organization, including those who install the systems, being internals or externals. The other cause is that the written policy is very detailed and cannot be implemented in many systems, at least not with the standard tools available.

Recommendations in this area are to:

1. Make sure the password policy is communicated thoroughly to everybody in the organization, and to all external suppliers and contractors.
2. Write a password policy that can be implemented on all systems where it is supposed to be implemented. This should be done in cooperation with various technical environments (operating system, application, database), as well as representatives for the helpdesk and normal users.

Creating user accounts and passwords

As I mentioned earlier easy passwords are often used when creating new accounts, sometimes with a parameter forcing the user to change the password upon his/her first logon. This introduces another risk. When 50 accounts are created with the same password, I have often seen several such accounts unused for several months, if they ever get into use at all.

The possibility to force the user into changing the password upon their first logon doesn't exist in relatively few systems, and it can be discovered and circumvented in some cases.

The recommendations here are to:

1. Ensure that good and unique passwords for each account are being set initially. This should normally be supported by the automatic generation of random passwords.
2. Regular audits should be performed to check for new accounts that hasn't been utilized after creation, and/or haven't

had the initial password changed.

Opening a locked-out account with a new password

Compared to the previous section, it is also important to ensure the use of good and unique passwords when accounts are being opened after an account lockout (forgotten password), and that the parameter to force the user into changing the password upon logon is set (if available).

The cause of this is that others should never know or have access to another user's password. This principle is important, and account administrators like all others, doesn't have a need to know other users passwords. In case of an incident unnecessary suspicion may be put on account administrators, since they without the proper security in place may know or have access to other users' passwords.

Using forced password change upon first logon

Among several possibilities, Windows has the possibility to force a user into changing his/her password upon the first logon. However unauthorized persons can find that this parameter has been set without logging in or changing the password. The problem here is that the actual temporary password can be found without locking the account or changing the password.

This is also one of the reasons for why I recommend unique passwords for every new account, as this reduces the possibility of unauthorized persons gaining access to large amounts of data and possibly also applications. This way any unauthorized person will need more time to break in successfully, thus increasing the chance of getting caught.

No option to change a password

Many systems, including both hardware and software, don't give the users an option for changing their password. Seeing this in relation to the user-administrators trend of using the same simple password for several accounts, the risk of unauthorized access increases.

In general I see the lack of not being able to change ones password as bad practice. However this is a flaw that we should, as good security professionals, encourage developers and software vendors to fix and make such an option available.

Generic accounts

By generic accounts I mean accounts that are not normally used by humans. I call these *service accounts*. Service accounts are being used by backup, antivirus, monitoring and other types of software that runs as *background services*. Users and administrators should always logon to systems with accounts that are traceable back to them (personal accounts), meaning that nobody should ever have the need for logging on to a system or domain using a service account.

Many security audits that I've done shows that such service accounts has rather easy-to-guess passwords. In some cases the software installation program is to blame (even blank passwords are used!), in other cases humans are to blame for weak service account passwords.

There is no reason for having it like this. On the contrary, service accounts should have passwords that are 'impossible' to guess or crack using brute force within a given timeframe. In my opinion that is a password with at least 15 characters in length, using both upper/lower case characters, numbers and special characters.

Given that such long passwords are difficult to remember they may be written down on a piece of paper, put into a sealed envelope that stored in a secure location with the system manager, operations security officer or similar in case of emergency. A log should be maintained for the use of such passwords, and the passwords should be changed after each use.

Deleting unused accounts

We are really good at creating new accounts. However we are not equally good at deleting old and unused user accounts, or even generic accounts (guests, testing, training etc..). Numerous security audits have shown me that in most systems there are open accounts which haven't been used since the system got installed, often several years ago. This same observation applies to user accounts belonging to people who have left the organization many years ago.

Regular controls should be focused on removing old accounts that are no longer in use. These use unnecessary resources, as well as representing a potential way into the system for unauthorized people. Although a drawback considering time and resources, I do recommend a simple check to verify the validity of the user and the users need for access before any account deletion. The easy way to do this check is of course to disable it, and wait for a reasonable time to see if it gets reopened again. If not, the account can be deleted. Factors such as vacation, absence due to sickness and business travelling must be counted in for the timeframe before deletion.

Changes to the policy takes a long time to live up to

Across most systems users will change their passwords at different times and intervals. This may be considered positive from a security point of view, but it also means that it takes a long time before the policy change has been made, and the point in time where all accounts are in compliance with the new policy.

A simple example:

Company X has a policy stating that all passwords must be a minimum of 7 characters in length, contain both letters and numbers, and the passwords must be changed every 60 days. The actual technically implemented policy is 6 character length, no requirement for upper/lower case letters, no requirement for numbers or special characters, using the username as whole or part of the username is allowed etc..

The company then decides to change their policy from minimum 7 to 8 characters in length, and no other changes. This requirement is implemented wherever possible, and it is the only this parameter gets technically changed/implemented.

First of all there's a timeslot of 60 days where some users will change their password to one with a minimum length of 8 characters. My experience has shown me that less than 50% of all normal user accounts actually changes their password within this timeframe. The exceptions are many: service accounts, employees on leave/vacation, users with the parameter 'password never expires' (there usually are many more than one will imagine), old and possibly inactive accounts (but not disabled), group accounts, test/training accounts and so on.

An upcoming whitepaper from me on auditing and improving Windows password security actually shows that even such a small change as above requires at least a year before 50% of all accounts can be considered to be in compliance with the new policy. Of course, active follow-ups and general housekeeping may reduce the time needed to implement a new policy, but in most circumstances it takes a lot more time than most people would initially believe.

By performing regular controls during such a timeframe we can generate statistics that shows the progression, clearly documenting the value of the security work being done.

The length and complexity of passwords

Passwords shouldn't be difficult to remember. On the other side they should be hard to guess for unauthorized people. Many people seem to believe that #=86cdgf%/&! is a good password, while 'My name is Per Thorsheim, and I live in Norway' is a bad password. On the contrary, I believe. The first password is very difficult to remember, and will probably be written down somewhere. More probably it will never be changed, as it is considered a really good password.

The other password is easy for me to remember. It's long (47 characters), it contains a word separator, upper/lowercase letters and a comma. In fact, technically this password will be almost 'impossible' to crack on Windows systems, for technical reasons that I won't explain in detail here.

Technical implementations of a password policy will usually set limits to the number of unsuccessful logon attempts one can do during a session or during a shorter period of time (say two minutes). Furthermore there may be a mechanism that disables an account after a given number of failed logon attempts. Last but not least there should also be a log that shows system administrators failed logon attempts.

One typical finding in a Windows environment is a policy that allows 3-7 logon attempts before the account gets locked, or at least X number of logon attempts every XX minutes without locking the account permanently.

Given the amount of information publicly available about me, I don't think that 'My name is Per Thorsheim, and I live in Norway' will be the first password guess of an unauthorized person. Let's say that this password will be number 200 of all passwords randomly guessed by an attacker. With 6 attempts every 10 minutes (or else the account gets permanently locked and becomes useless) one can do 36 passwords per hour, and it will take almost 6 hours to actually get access to my account and password, when the process is automated and runs continuously.

I really hope and believe that physical security, logging, intrusion detection mechanisms as well as due care will discover these logon attempts within such a time period. But remember; an unauthorized person may try out a very small number of passwords across multiple accounts to gain access, instead of focusing on just one single account. This is just as easy to automate, and is usually more successful in a fraction of the time required for the first type of attack.

I believe that it is time to increase the password length and –logic instead of using short and complex passwords to achieve better password quality (changing from passwords to *pass phrases*, as it is commonly referred to). Last but not least we have to make an extra effort to harmonize our written and technically implemented password policies, so that users doesn't have to relate to different sets of password rules across different systems.

Most organizations have got a single written password policy. Why then do they have tens or even hundreds of different technical implementations of it?

About the Author

The author is currently finishing a large whitepaper entitled 'Auditing and improving Windows password security, which will be made available at www.thorsheim.net and www.passwordaudit.net during summer 2005, along with tools, tutorials and other materials.

Due for release during summer 2005. Please check my website www.passwordaudit.net for the latest information on this.