

## Passwords – The rules and the reasons

By David Inglis, Protiviti KnowledgeLeader Senior Content Developer

### Introduction

Passwords are a prime source of irritation to all who have anything to do with them. As users, often we're not allowed to chose short passwords, we're forced to change them every so often, we're supposed to keep them secret, remember several at a time, and not use the same password for everything. As administrators, we have to deal with people forgetting passwords, getting locked out of systems because they typed the wrong password several times, or sharing their password with other people. Because passwords cause so many problems, why do we use them?

### What is the purpose of a password?

Passwords are required when requesting access to a computer system during some kind of login process. Some of the more common uses include requesting access to: user level accounts, web accounts, email accounts, screen saver protection, voicemail access, and local router logins.

Passwords are used to authenticate users, as part of a process that today is becoming known as Identity Management. When a user logs in to a computer system, the system needs information to perform two tasks.

1. Identify the user: The system has to uniquely identify the person logging in, in order to determine not only what that person is allowed to do, but also what the person is not allowed to do.
2. Authenticate the user: The system needs proof that the person logging in really is who they claim to be. This is very important, since once someone is successfully logged in to a computer system, then so far as the system is concerned, that someone **IS** the person associated with the identification information, and gains all the rights, privileges, and access of that person.

The first piece of information is usually called the 'Username', 'Accountname', or 'User ID', and is typically comprised of alphanumeric characters forming something that either:

- Relates to the user, for example some form of the person's name; or

- Relates to the system or application being accessed, for example an email address.

This piece of information is usually not secret, or at least may easily be constructed from simple rules that allow someone to determine the 'Username' associated with a particular person. In most cases, the Username is generated either by a computer administrator or by the system, and is typically known to several people. It is because the Username is typically known to (or can be easily determined by) several people that some other information is required to 'authenticate' the user. This extra information can be one or more of the following:

- Something the user knows: A word, phrase, date, name, character sequence, etc. that only the person logging in should know;
- Something the user has: A physical object such as a key, 'smart card,' or 'token;'
- Something the user is: A fingerprint, voiceprint, retina pattern, etc.

A password is an example of the first of these: something the user knows. However, it is only of use for authentication if it is also something else: a secret. It is therefore important to take steps to prevent passwords either being shared, being created in a way that makes them easily guessed or constructed, or being stored in such a way that they can be easily read.

### **Why use passwords for authentication?**

Using a password (something a user knows) is cheap, easy, and flexible. Passwords are usually entered into a computer system through the same interface that is used for almost everything else: the keyboard. Occasionally the interface might be a touch screen or just a numeric keypad, but the point is that no additional hardware is needed to enter a password.

Contrast this with movie scenes in which people speak a phrase, or look into a camera, or place their hand on a 'reader' of some kind before being allowed access to a facility or computer system. All of these biometric identification mechanisms (something the user is) are currently in use as authentication mechanisms, but generally require expensive hardware or software to work well, and often also need a significant amount of 'training' before they will recognize people with an acceptable degree of reliability. In most cases, they are simply not yet 'ready for prime time.'

What about something the user has? Keys have been used for thousands of years, and in a modern form (i.e. an ID badge with a transponder inside) are used every day to control access to different parts of a building. Even an ATM card can be thought of as a key, since inserting an ATM card provides access to a facility (an on-screen menu) that is not otherwise available. However, very few general-

purpose computer systems have card readers attached to them, and in these situations tokens may be used instead.

Essentially a token is a 'smart card' that displays a pseudo-random password that changes at regular intervals. When requested by the system the user enters the password currently displayed by the token, and the system then compares that password with one it has obtained from a 'master' device that generates the same pseudo-random password at exactly the same time. This then proves that the user has access to a valid token. This system is used in many organizations, but has one major flaw – the token can be lost or stolen. In addition, extra hardware and software is needed on the server. Finally, cards or tokens have to be issued to every user, which typically restricts their use to people within a single organization.

The bottom line is that very few computer systems have support for authentication using a physical object (something you have) or biometrics (something you are), and this includes all the systems with which people are most familiar. Even where physical objects are used (e.g. ATM and credit cards), they are not used for authentication. Instead, swiping a card through a reader just provides a convenient way of inputting an account number (the 'Username'). It is the PIN (Personnel Identification Number) that provides the authentication. Because passwords (or PINs) are relied upon so much to grant or deny access to such a vast amount of important information, every computer user should know how to select and use passwords that will not become known to other people - strong passwords.

### **Password Strength**

Passwords that are difficult to guess or otherwise determine are termed 'strong passwords', whereas passwords that are easily guessed or determined are termed 'weak passwords.' The strength of a password is a measure of the amount of effort (either manual or automatic) required to identify it, and is determined by three characteristics:

1. The length (number of characters) of the password;
2. The number of different symbols (letters, numbers, etc.) allowed in each character position,
3. The 'randomness' of the actual characters used.

The greater the degree of each of these characteristics, the stronger the password will be. For example, a password consisting of only three characters, each of which can be any of the numbers 0-9, can have any one of  $10 \times 10 \times 10 = 10^3 = 1,000$  different combinations. On average, someone trying to guess the correct combination would find it after only 500 tries, and so it is easy to see that such a password would be very vulnerable to a 'brute force' attack, in which every combination is tried in turn. Even trying this manually is perfectly feasible, although tedious. However, it would present no problem at all to an automated

'password guesser' program, and consequently a password of this form would be considered very weak.

By increasing the length of the password to four characters (like many PINs), the number of possible combinations increases to  $10^4 = 10,000$ . Only a very determined person might be prepared to try an average of 5,000 times to guess a four-character password, but this would still cause an automated system no problem. If the length increases further to six characters, the number of possible combinations increases to  $10^6 = 1,000,000$ . Then, if each of the characters is allowed to be any of the letters a-z instead of the numbers 0-9, the number of possible combinations becomes  $26^6 = 308,915,776$ . Although this number of combinations would easily defeat a manual attack, it would still present little problem for a password guesser program.

However, if the length of the password is further increased to eight characters, and each of the characters is allowed to be any of the symbols a-z, A-Z, 0-9, or any of the 30 punctuation characters found on a typical English language keyboard (!@#\$%^&\*()\_+{}|:~<>?=-[]\;',./), then the number of possible combinations becomes  $(26+26+10+30)^8 = 92^8 = 5,132,188,731,375,616$ . Even at a rate of one guess every microsecond (i.e. 1,000,000 per second), it would take an average of over 162.7 years to guess a password having this number of possible combinations, making this (potentially) a very strong password.

## Choosing Strong Passwords

Even requiring that passwords meet the 'strong' characteristics given above does not guarantee that these passwords will be strong in practice. For example, if someone's username was [Eric@1stStreet](#), then [Eric@1stStreet](#) would represent an extremely weak password for that particular person. In addition to the above considerations of length and allowed characters, to be truly strong a password must not be any of the following:

- A word in any language, slang, dialect, jargon, etc.;
- Birth dates, or personal information such as addresses and phone numbers;
- Other personal information such as names of family, pets, friends, co-workers, fantasy characters, favorite sports teams, etc.;
- Common computer terms, names, commands, sites, companies, hardware, software;
- The name or location of the company the user works for, or any local place names;
- Word or number patterns such as aaaabbbb, qwertyui, zyxwvuts, or 12344321;
- Any of the above spelled backwards;
- Any of the above preceded or followed by a single digit (e.g., secret1, 1secret).

The above list is not meant to be exhaustive!

In general strong passwords have the following characteristics:

- Contain both upper and lower case characters (i.e., a-z, A-Z);
- Contain numbers and punctuation characters as well as letters;
- Are at least eight characters long;
- Are not any of the words in the above list.

In contrast, weak passwords:

- Contain fewer than eight characters;
- Do not contain both upper and lower case letters;
- Do not contain either numbers or punctuation;
- May be a word from the above list.

## **Passwords, Passphrases, and PINs**

Passwords and passphrases are not exactly the same thing, even though they are constructed from the same elements. Although a phrase may be used as a password, the real reason for using a passphrase is to make it possible to have enough characters (at least more than 20, and possibly more than 30) to use as the starting point (the 'seed') in an encryption process. Because it's not easy (!) to remember 30 (or even 20) random characters, a phrase has to be used to help.

As a phrase is generally significantly longer than a single word, selecting a phrase such as "BeamMeUpScottie" (15 upper and lower case letters) might seem like a good idea for general-purpose use. However, beware! A book of quotes might contain 40,000 quotes or more, and so any common phrase will almost certainly be guessed very quickly by a program that has been loaded with books of quotes. To be safe, any phrase found in any published form should not be used. However, "ScottieMeUpBeam" is better, and "5cott1eM3UpbeaM" is better still. In general, when using a passphrase, follow these simple rules:

- Re-order the words;
- Swap some near-equivalent letters and numbers (0-O, 1-L, 2-Z, 3-E, and 5-S);
- Change some words into 'number equivalents', e.g. use "4" instead of "for;"
- Randomly change the capitalization.

Strictly speaking, PINs and passwords are different things. PINs contain only numbers, and they may contain as few as four (very weak, but easily remembered), or may be much longer (in which case they may be stored in a magnetic stripe on a card). PINs are often assigned by someone else and are then used for identification, not authorization. Of course, it is perfectly possible to use a password that consists purely of numbers, but that doesn't necessarily make it a PIN.

## **Password Creation and Storage**

Even when a strong password has been created, misuse of that password can easily lead to it being lost or stolen. Therefore, a number of rules are normally

applied to the ways in which passwords are created, stored, and used, to try to ensure that only the person who created the password ever knows what it is.

### **Password Creation**

The first thing to note is that it should never be possible to view a password on a computer screen, in order to prevent 'shoulder surfing' (literally where someone looks over your shoulder and notes what is on the screen). Instead, typically each character that is typed shows up on the screen as an asterisk (\*).

Because of this security feature, it is impossible even for the person entering a password to be sure after the event exactly what they typed. Therefore, whenever a new password is created, to be sure that the person entered what they thought they did, the password must be verified by being entered into the system for a second time.

### **Password Storage**

Once a password is entered, it must be stored in such a way that it cannot be read by anyone looking at the file where the password is stored. Typically, all passwords used for access to a particular system are stored in the same password file, and the name and location of this file is usually common knowledge among hackers. For this reason, not only must the file itself be subject to stringent access controls (so that it cannot be read or copied without system or security administrator access), but also the passwords must be stored in such a way that even administrators cannot read them.

In order to prevent passwords from being read, all passwords are encrypted when entered, and all stored and transmitted copies of passwords are in this encrypted form. The passwords are encrypted using what is known as a one-way function, which means that it is impossible (strictly speaking, it is computationally unfeasible) to decrypt passwords once they have been encrypted. This also means that passwords cannot be decrypted for comparison purposes when a user attempts to log in. Instead, the new password typed by the user is encrypted using the same one-way function, and the two encrypted copies are compared.

### **Dictionary Attack**

Encrypted passwords stored in a password file can be guessed through what is termed a dictionary attack. This relies on the fact that the one-way function used to encrypt the password is usually not secret. As a result, a hacker can choose a word, encrypt it, and then see if it matches any encrypted passwords in the file. If a match is found, then the hacker has found the password. Typically, a hacker does not just guess or make up words. Instead, a dictionary file containing many thousands of words is used, and each word is tried in turn. It is for this reason that a password should never be a word found in any dictionary.

## **Password Controls**

In addition to the requirement to create and store passwords securely, they must be used securely, and consequently there are a number of controls that are placed on password usage.

### **Password Expiry**

Perhaps the most common complaint by users regarding passwords is that they cannot use the same password for as long as they want. Instead, they are forced to change their passwords after some time period (usually a number of months).

If a password is in use for a short time, then it is unlikely that anyone other than the creator of the password will find out what it is. Conversely, the longer a password is in use, the more chance there is that it will be guessed or divulged to someone else, either accidentally or on purpose (more dictionary attacks, more guesses based on something about the person, etc.). Therefore, all passwords should be changed after they have been in existence for some specified period of time, and this is achieved by expiring them. An expired password cannot be used for the purposes of logging on, although it is still stored by the system for some time after expiry.

### **Password Lifetime**

The lifetime of a password is the length of time it can be used before it expires. In most low-risk situations (i.e. for accounts that do not have special privileges, and where only 'low value' information is accessible), passwords expire after 90 days (or 180 days in very low-risk situations). When data is valuable, or when accounts have privileges (e.g. systems administrator accounts), passwords typically expire after 30 days.

Most systems provide a 'count down' before a password is expired (for example, during the last week), thus allowing the user to change to a new password before expiry. If an account is in regular use, then a user typically has plenty of warning and can create a new password before being locked-out of the account. However, where an account is not in regular use the user may only find out that the password has expired after the event.

### **Password Re-use**

A new password is not allowed to be the same as the one being replaced, since creating a new password identical to the previous one would have no beneficial effect (i.e. it would result in no net change). For the same reason, it must not be possible to simply perform a 'flip-flop', i.e. to change the password twice in quick succession, ending up with the original password. This restriction can be achieved in one of three different ways:

- By storing the most recent passwords created by a user (typically the last 5 or 10), and not allowing that user to re-use a password that is currently stored;

- By setting a minimum password lifetime: This prevents the current password from being changed until after a certain minimum time (typically one day) has elapsed;
- By setting a minimum password re-use time: This prevents a password from being re-used until after a certain time (typically one year) has elapsed.

The first two of these controls are usually implemented together, as implementing one without the other provides virtually no increase in security. However, as the first two controls together can easily be defeated within a few days by a really determined user, for the highest security a minimum password re-use time must be set. The downside of this is that all passwords must be stored for a year after they have expired or been changed.

### **Forgotten or Expired Passwords**

In most organizations the most common reason for a user to contact a Help Desk or Call Center is to obtain a new password, either because an existing password has been forgotten, or because the password has expired. In either situation the Help Desk then has to assign a new password to the user, and this password has to break some of the above 'rules':

- The password is known to at least two people – the person who created it, and the user who then uses it to log in;
- The password has to be transmitted to the user in 'cleartext'. In other words, the password is not encrypted.

For these reasons these 'special' passwords are subject to some additional rules to reduce the possibility of them being stolen:

- The passwords can only be used once, after which they expire;
- The passwords are usually only valid for a short time period (possibly just a few hours).

The other problem associated with passwords assigned by a Help Desk is one of user authentication. How does the Help Desk know who the person calling for a password really is? The answer is – they don't know, and this form of deception, also known as 'social engineering,' is a very common way of obtaining unauthorized access to a system.

If someone can obtain the name and username of a user, that person can pretend to be an authorized user and can ask for a new password. The solution is for the Help desk to refuse to give out new passwords over the phone, but instead insist on using an already-established delivery mechanism, most often the authorized users' email address.

### **Three Strikes and You're Out**

Password guesser programs use various different methods to try to determine a users' password:

- Manipulation of the username (backwards, swapping characters, changing capitalization, etc.);
- Trying words from one or more dictionaries;
- Brute force (i.e. trying every possible combination in turn).

It is not at all difficult for a hacker to create a program that attempts thousands of logins a second, using different passwords each time, until the correct password is found. As a result, many systems impose one of the following restrictions to defeat such programs:

- Only a limited number of consecutive failed login attempts (typically three) are allowed within a certain time period;
- Only a limited number of consecutive failed login attempts are allowed;

The first case allows someone to try to determine a password an indefinite number of times, but only at a rate below the threshold. This may not actually stop a password guesser, but instead forces it to run so slowly that it is pointless. It also runs the risk that the failed attempts will be reported the next time the authorized user logs in. The second case is more restrictive, in that there is no time limit – it is only the number of consecutive failed attempts that counts.

In both cases the end result is the same. Once the limit is exceeded, the account is locked and cannot be used by anyone, even the authorized user. How this is dealt with then depends on what is trying to be achieved:

- If all that is required is to slow down password guesser programs so they become pointless, then the account can be automatically unlocked after a specified time period (e.g. one hour).
- If an investigation of the (possible) break-in attempt is required before allowing access to the account, then the account is only unlocked by action of the Help Desk..

## **Stolen Passwords**

How can a user tell that their password has been stolen? If the person who stole it never attempts to change it, then possibly the user may never know that someone else is using his or her account. However, there is one very simple thing that can be done to help users spot unauthorized access. On some systems the following messages are output on the screen immediately after every successful login:

- A message reporting how many unsuccessful login attempts there have been since the last successful one;
- A message reporting the date and time of the last successful login.

Users can then compare this information with their access pattern, and determine whether there have been any login attempts (successful or otherwise) since their last login. An unsuccessful attempt by someone else is a clear warning that someone is interested the account. The security administrator should be notified immediately, and a weak password should be replaced by a strong one. If the

account has been used by someone else but the authorized user can still login, then again the password must be changed, but in addition the security of the whole system is suspect and security personnel need to perform thorough checks.

If a user finds that they can't login when using the correct password, and the Help Desk does not indicate that there is any system problem, then someone else may have changed the password. If the system records when passwords were changed (as it should), then this may help to determine how this happened. Even if this information is not recorded, the security of the whole system is suspect and security personnel need to perform thorough checks.

### **Letting Other Users Have a Password**

This is not really the same as having someone steal a password, but the result may be the same – an unauthorized user gains access to a system. Here, the authorized user may well not be concerned with this (and may not consider it to be a problem), and so the controls described above may operate in a slightly different manner:

- Password strength: The stronger the password, the more difficult it is likely to be for someone else to be able to remember it, particularly if the password is not associated with any particular memory or thought process of the other person;
- Password expiry: Every time a password expires, or is changed for any other reason, then all other people who have been told the old password will lose access unless they are told the new password too;
- Forgotten or expired passwords: Only the authorized user will receive a new password from the Help Desk;
- Three strikes and you're out: The more people who know the password, the more who are likely to forget it, increasing the probability that the authorized user will be locked out.

All the controls described, while not preventing anyone from deliberately sharing a password, make it more difficult to maintain the sharing, and also create more work for the authorized user.

### **Remembering Passwords**

One of the 'Golden Rules' regarding passwords is that they should never be stored anywhere in cleartext – and this includes not writing them down on any pieces of paper, inside any books, on post-it notes, or anything else! Nor should they already exist in written form, for example on a calendar, poster, book or magazine cover, or anything else that is easily visible. This means that passwords should be easily remembered, but of course an easily remembered password may then be an easily guessed password.

Own way round this is to use something that is so personal that no-one else might possibly guess it (for example, the name of the driver who took you on that

particularly memorable trip), or to create a passphrase that has some chain of association that is not meaningful to anyone else). Like most rules, even the 'Golden Rule' described above has an exception. This is where a written copy of a password is kept in a secure place, for example a locked safe, for the purpose of allowing emergency access under controlled conditions by a designated person.

## **Conclusion**

Most people complain about the password rules and restrictions they must follow. Either passwords are too long, or have to include numbers or punctuation, or have to be changed too frequently, etc. Whatever the complaint, there is a reason behind the rule, and ultimately that reason has to do with preventing unauthorized access to a system. Without these controls, passwords are simply an ineffective means authenticating users, thus defeating their purpose entirely.

*This white paper was contributed by KnowledgeLeader, a website providing tools, templates, and other resources for internal audit and risk management professionals. The KnowledgeLeader Internal Audit and Risk Management Community is designed to help companies stay up-to-date on current internal audit, business and technology risk issues, and to become more efficient and effective. The service is provided by Protiviti, an independent firm dedicated exclusively to risk consulting and internal audit.*

For more information, see <http://www.knowledgeleader.com/>.

For questions please telephone: 1 866 923 8513. For requests from outside the U.S. or Canada, call +1 925 598 7771.

Material from the KnowledgeLeader® Internal Audit and Risk Management Community

<http://www.knowledgeleader.com>

© 2003 Protiviti. All right reserved.