Keeping Secrets: Good Password Practice

Everyone knows that passwords are important, but how do you keep them safe?

David Harley, BA CISSP FBCS CITP
Director of Malware Intelligence

Randy Abrams
Director of Technical Education

Table of Contents

Introduction

In 1997, David Harley gave a presentation on social engineering at a European security conference, that included a section on improving password management in the workplace. In the question-and-answer session afterward, the discussion largely ignored social engineering in general and veered into a debate as to why any businesses would still be using static passwords to manage their access control needs when better alternatives were becoming available? Some 12 years later, alternative and supplementary forms of authentication have become far more common (and affordable), but the humble password remains the somewhat crumbling gatehouse to many a security structure.

The X-Factor

Passwords and passphrases are one example of a form of security measure called authentication. We give them to people so that they can use them to prove that they are entitled to use something to which access is restricted: for example, a restricted area, a computer system, an Internet service or an encrypted document. An authentication factor is a procedure or chunk of information used to authenticate an individual, that is, to verify that he or she has access rights. Authentication factors generally fall into one of three classes:

- Inherence (or human) factors — "Something you are (or do)." These include biometric measures for identifying the individual by a physical characteristic such as fingerprints, iris or retina recognition, or automated face recognition, or alternatively by behavioral characteristics such as typing rhythm or voice mannerisms.
- Ownership (or technical) factors — "Something you have." These include identity cards, communications devices such as passcode tokens, or even a traditional metal key.
- Knowledge (or personal) factors — "Something you know." These include passwords or passphrases, a PIN (personal identification number) or other passcode, the entry combination for a door unlocked by keypad, the answers to a series of questions (challenge-response) and so on.

Where more than one authentication factor is used, this is referred to as two- or three-factor (or even multi-factor) authentication.

- A "chip and pin" bank card, which requires both the possession of the card and the knowledge of the PIN that goes with it, is an example of two-factor authentication commonly met with outside the U.S.
- Hardware tokens are often implemented as a two-factor measure, since the user needs both possession of the token and knowledge of a PIN in order to use it.

- The addition of other measures such as a username/password combination to access a specific device or service, or a biometric device such as a fingerprint scanner built into a mouse or laptop, increases the authentication factor count, and hopefully the security of the access control system.

The Kryptonite Factor

Any single factor has its strengths and weaknesses. In the case of a passphrase, the main risks are that the phrase will be guessed, intercepted (electronically as by "sniffing" an unencrypted transmission, or by discovering a written copy), or shared inappropriately by a legitimate holder (for instance, when tricked into divulging it by a social engineering attack).

The sad fact is, static passwords are a superficially cheap but conceptually unsatisfactory solution to a very difficult problem, especially if they aren't protected by supplementary techniques such as password ageing (enforced password changes after a period of time specified by the administrator) and restricting the number of failed password attempts allowed. It can be convincingly argued that the apparent cheapness of the solution is often illusory, being outweighed by the security risks present in a poor implementation and the hidden administrative overhead. Many sites that are dependent on static passwording have helpdesks drowning in trouble tickets relating to forgotten passwords. One-time passwords are much more secure, especially when implemented in hardware as a two-factor authentication measure, but have other disadvantages — chiefly expense in implementation and the risk of inconvenience when a device is lost or loses synchronization with a central server and so on. Effective biometric services devices are often expensive to install and maintain.

So Why Do Good Passwords Still Matter?

- One of the most common forms of automated attack on a corporate system is password guessing, while attempting to steal passwords from targeted individuals by various forms of social engineering such as spear phishing is something of a growth industry.
- On many systems, most untrusted services are still protected primarily by passwords rather than by more glamorous methodologies such as smart cards, biometric systems, hand-held authentication and so on.
- Insecure passwording, whether as a result of bad systems practice or bad user practice, may endanger data in breach of data protection legislation, contractual obligation, or corporate policy. Note that this risk exists whether or not a poor password is *actually* stolen or exploited. Just the existence of an unexploited loophole may be considered a breach.

Good User Practice

It has been estimated by CERT-CC that "80 percent of all network security problems (are) generated by bad passwords."[1] We can't actually demonstrate the accuracy of that statistic, but it does at least indicate anecdotally how seriously the security community regards the issue of password security. So here are some ideas that should make you less vulnerable to password-related problems as an end-user.

Sharing Is for Candies, Not Passwords

Don't share passwords unless there's a formal protocol set up to allow it. More than one person sharing an account is (except under "very" controlled conditions) a major threat to security.

At the very least, it presents difficulties in tracking password-related problems, even where no malicious intent is suspected. Unless clearance in writing has been obtained from an appropriate person, password sharing may be regarded as a breach of discipline. Note that in many cases, an "appropriate person" is not the head of a client unit (marketing, human resources, legal department) but the system manager or equivalent for a password-restricted person, perhaps even someone higher in the food chain, such as the IT director.

The dangers of password sharing are not restricted to vulnerability to malicious attack. The integrity of shared data can be compromised in a number of ways (overwriting by incorrect versions, inadequate file or record locking, accidental deletion) unless sharing is properly organized. However, the possibility of malicious attack introduces a number of complications:

- The more people with access, the greater the risk of accidental or deliberate extension of access to intruders.
- The more people with access, the easier it's likely to be for someone to crack, guess, or use social engineering to gain the password illicitly.
- Any breach of security on one networked computer is likely to compromise security on the whole of the subdomain, perhaps the whole corporate network.
- Attacks on computer systems can come from inside as well as outside. This should be of particular concern to an end-user who is not cautious about sharing account information. If an attack is traced to a particular account, the holder of that account will be the prime suspect.

Anti-social Engineering

Be aware of the social engineering approach to cracking passwords. The quickest route to appropriating a password (especially a shared one) can be via a phone call and a bluff. Don't disclose passwords to anyone whose identity you can't verify, or whose right/need to know is in doubt.

Beware of any request for passwords and PINs, especially if received by email or in a phone call that you did not initiate and can't verify independently. This is a particular issue with phishing attacks, and techniques for recognizing phishing attacks have been considered at length in other ESET papers.[2]

It's very easy to change the apparent point of origin of phone calls and emails (also other forms of messaging). If someone rings you about a problem with your bank account, email account and so on, it's a good idea to verify his or her identity before giving out any sensitive and/or identifying information. It's better to contact them back via a confirmed "good" number or email account rather than via a contact point that they give to you. Clearly, the fact that they gave it to you proves nothing about its authenticity.

A good rule of thumb is that anyone who contacts you in order to ask for your credentials should be treated with suspicion. However convincing their story is, verify that they are who they say they are. A systems administrator should never need to know your password in order to maintain a system to which you have access. If a helpdesk operator or engineer asks you for that sort of information, the chances are that they're either incompetent or not who they say they are.

Trojans and Tricksters

It's not only people who masquerade as something they're not. Trojan horses take many forms, and many of them are intended to steal passwords and other sensitive data. Treat any uninvited program with suspicion, from whatever apparent source. A web site or similar resource can also be a sort of Trojan horse. Sometimes a legitimate site is compromised with malware, and sometimes a criminal goes to enormous lengths to make a spoofed or otherwise malicious site look legitimate. If you're told you need to access a given resource for some convincing reason, verify the link independently to ensure that you're not accessing a fake site set up specifically to capture your account name and password.

It isn't only financial transactions you need to worry about; gaming sites and social networking sites also provide a rich habitat for criminal exploitation. Even if your own account on Facebook, for example, isn't important to you, bear in mind that it might be used to defraud others, and such criminality may have unexpected and negative consequences for you, too.

Password Selection Strategies

It's good practice to avoid the following common strategies when deciding on a password.[3] We're referring here to using these approaches for single, fairly short passwords. Some of them are acceptable in combination with obfuscation techniques such as interleaving one word with another and using the resulting token (or string of characters) to form one element of a long passphrase, but we'll get back to that shortly. These techniques are intended to reduce the risk of your password or passphrase being discovered either by guesswork from someone who knows something about you, or by an automated dictionary attack, where software is used to go through a long list of words and character combinations, trying each one as a password. Some Trojans use comparatively short, generic lists of commonly used passwords such as "aaaaa," "password," "qwertyuiop," "StarTrek," "mypassword," "123456." If you don't believe that such stereotypical passwords represent a significant problem, check out "The Top 500 Worst Passwords of All Time" at http://www.whatsmypass.com/?p=415. We can't confirm the exact ranking, but we've certainly seen very many of these used in real life. Table 1 shows the top 10, according to the site.

Table 1: The 10 Most-Used Passwords

1	123456
2	password
3	12345678
4	1234
5	pussy
6	12345
7	dragon
8	qwerty
9	696969
10	mustang

(Source: http://www.whatsmypass.com)

At the other extreme, a dictionary attack may use not only common "strings" of characters like these but lists of hundreds of thousands of real words. This may strike you as being a little over the top for capturing your Twitter credentials. However, modern computer systems are fast enough to carry out an automated attack like this far more quickly than you might think.

Here are some approaches best avoided:

- Any correctly spelled English word, especially one which is likely to be recognized by operating system or application spell-checkers and so on. Using regional spellings, such as those from the UK, is unlikely to offer any extra protection.

- Any correctly spelled non-English word; exceptions may be a little more acceptable in obscure languages as long as they're not in languages you're "known" to speak, but you are still at risk from dictionary attacks that use long, multi-language word lists.
- Any part of your own name or username, let alone a duplication of your username (this is called a "Joe" account, and it's one of the first things a password cracker (human or automated) looks for when it comes to trying to guess a password).
- Any part of the name of a member of your extended family (including pets) or, worse, a colleague, your boss, or, in fact, anyone's name. Place names are often easily guessed, whether because of an obvious link to you (if you live in Springfield, Springfield is definitely not a good password choice, for instance), or because word lists used in dictionary attacks are likely to contain common place names.
- The name of the operating system you're using (or accessing remotely), or the name of the PC you're using, or the name of the service you're accessing, or the hostname of a server you're accessing. Well, you get the idea.
- Personally significant numbers (phone number, car license number, National Insurance or Social Security Number, someone's birthdate — save them for picking lottery numbers).
- Your favorite or most-hated objects, food, movies, TV programs.
- Easy associations with favorite or most-hated things; for instance, "Swan_Lake" may not be a good password for a ballet fan.
- Song, book and movie titles, famous people, cartoon characters, etc. Particularly not recommended are 'CharlieBrown,' 'Snoopy,' 'Kirk,' 'Spock,' 'Homer,' 'Garfield,' 'Dilbert,' 'Grissom,' 'Oprah'...
- Anything so unmemorable you have to write it down, unless you take reasonable precautions to protect the paper you write it on.
  - ☐ A Post-It on your keyboard or monitor is not a reasonable precaution, unless you work in a room that can't be accessed by other people.
  - ☐ Nor is a piece of paper taped to the CD or USB device it's intended to give access to.
  - ☐ A piece of paper in your wallet or laptop bag is vulnerable to loss or theft. At the very least, take measures to avoid its being easily identified as a password, and don't make it obvious which system/file/account it refers to. Don't write down the actual password; use a mnemonic device or some means of disguising it such as scrambling and interleaving letters.[3]
- Anything that is all uppercase or lowercase (unless the system is case insensitive!).
- Anything with the first or last character uppercase and the rest lowercase, unless it's a really tricky passphrase.

- Any example passphrase you've come across as in a textbook or an ESET white paper or blog.
- Any short passphrase consisting of a single word (system permitting — some systems actually severely limit the range of characters you can use).
- Anything consisting entirely of letters of the alphabet (system permitting).
- Obvious anagrams of any of the above, especially simple reversals.
- Obvious variations such as appending or prepending a digit to one of the above or an anagram thereof, or obvious substitutions of digits for letters: "pa55w0rd," for example.
- Reusing passwords can be really bad news. You don't want to use the same password for your computer logon as for your bank. Important information should be protected with unique and strong passwords.

Filling the Cracks

Techniques that may help in slowing down password breaking by guessing or simple dictionary attacks include the following.[4] The more combinations of techniques you use in a single password, the more effective they're likely to be. However, sophisticated crack programs will attempt to counter many of these strategies.

- Embed control characters or non-alphanumeric symbols such as digits, punctuation marks and symbols (where the system allows this).
- Misspell (but consistently!) "Dis passéfrase 1s kwite gud bot wd b betr wiv sum #s & karakters that r nut alfan00meric."
- Unorthodox caPitaliZation
- Use a personally significant acronym, e.g., ICRMFPW (I Can't Remember My Friendly Password)
- Link together two words, possibly with a symbol as a delimiter, e.g., egG^rIbBoN.
- Replace letters with digits or equivalent characters, and words with abbreviations, e.g., BunZ4T, NeWz@10.
- Interleave two words, e.g., RmAaInN.
- Interleave a word with a numeric string, e.g., f9L7a0s8H.
- Don't use the same password on several machines. However, sensible variations might be acceptable, subject to the rules mentioned above, e.g., VdOOmAX, UdOOMniX, dOOCPM. Still, this example has the disadvantage that if an attacker gets one of these, he's well on the way to guessing the rest.

- Changing your password regularly is important. How frequently you change your password will depend upon how important the information you are protecting is. Generally, once every three months is a really good idea. That way, by the time a computer has cracked a good strong password, you will have already changed it!
- One of the problems with multiple passwords is remembering them all. Tools like Cygnus Password Corral (http://cygnusproductions.com/freeware/pc.asp), Keepass (http://keepass.info/), and 1Password (http://agilewebsolutions.com/products/1Password) can be really helpful. Just remember that you need to keep your "keysafe" application on a very safe computer, and back up that password file!!!

Case Study: The Happy Twitter Admin

According to the media, a recent Twitter hack was made easy because an administrator used the password "happiness" and made it easy for a hacker to gain access to an account. This password is all lowercase letters, which means it can be brute force cracked with only a pool of 26 characters. In practice, most cracking tools will use at least 52 characters, so as to include both upper- and lowercase. Since the password was a word, a more efficient dictionary attack using a word list cracked it quite quickly. At nine characters long, there were only about 5.4 trillion combinations that the password could possibly be. However, since the password is an English word, this reduces the possibilities to about 1 million. A single word, no matter how obscure, is a terribly weak password. If the password had been "happinessis," then it would not be a word, it would still be easy to remember, and there would have been about 3,670 trillion possible combinations of lowercase letters for an 11 character password. This is more than 650 times better than any nine-character lowercase password.

With a nine-character password that contains upper- and lowercase letters, numbers, and special characters, there are almost 630,250 trillion combinations. Now for something really interesting. If the admin had used the password "happinessisgood," there would have been more than 1.6 billion trillion combinations possible, using only lowercase letters. Yes, 1,677,259,342,285,730,000,000 possibilities with a 15-character lowercase password. That means it takes a long time to crack the password. This also means that a 15-character, all-lowercase password, that is not a single word, is much stronger than any nine-character password, no matter what special characters you use! Is it really that much harder to remember "happinessisgood" than "happiness"? Simply changing the password to "Happinessis2good!" makes an incredibly strong password.

Remember, size does matter more than complexity, as long as you do not use just one long word. The longest word in the English language is still only one in about a million words and very easy for a computer to guess in a short time.

While we've tried to discourage the use of easily remembered (but easily guessed) words or character strings, like your dog's name or your birthday, you can still use these safely if you are smart about it. "Rover loves 2 run" is a fine password. "On 4/17/60 I entered the world" is a very strong password that contains somebody's birthday!

The administrator at Twitter would have found much more happiness and security using a simple password with only a tiny bit more complexity and a few more characters, and so will end-users!

Despite the IT policies that are prevalent throughout the world, really great passwords can be created that do not use upper- and lowercase letters with numbers and special characters. The really important thing is length. Actually, "The really important thing is length" is a much better password than $kW3P*v9.

There are several reasons why the sentence above is a better password. To begin with, you can remember it so you don't have to write it down and keep it handy. Even more importantly, it will take a computer far longer to crack the sentence (unless the password-cracking program knows it needs to look for a sentence) than the eight-character password with all of the funny characters and so on.

Adding numbers and special characters does help, but, in general, not as much as length does. There is a time when the special characters do become important. That is when you are limited to a short password because of system or site policy. In a case where the system only permits a short password, you want uppercase letters, lowercase letters, numbers, and special characters, if the system allows.

One of Randy's favorite tricks for creating passwords that he can easily remember, and are nice and secure, is to make a math equation. Something like "1hundred+5=Threehundred" is long enough to be secure, has a nice mix of characters, and the wrong answer is silly enough to be memorable!

Notes for Systems Administrators

Password enforcement is a trade-off between paranoia and practicality. The tighter the restrictions, the likelier it is that the user will evade restrictions as far as possible so they can get on with the job reasonably conveniently.

- Be aware of technical vulnerabilities that may allow interception of passwords for specific systems.
- Generating good passwords for users and forcing them to use them, rather than choosing their own, has obvious advantages. However, it should be borne in mind that the harder they are to remember, the likelier the user is to write them down and keep them somewhere unsafe. Long, random strings of characters are not going to be memorized.

Mind you, in the world of encryption, things are a bit more complicated than using the longest password possible. It's possible to enforce really, really long passphrases and still have poor security because the encryption algorithm is poor or broken. However, in this paper, we're assuming that you're looking for ways of choosing good passwords, not designing secure systems.

- Unlimited attempts to access a passworded system should not be allowed.
- Where a login attempt threshold has been set, breaches should be audited. The user should be notified at the next successful login and encouraged to report anomalies.
- One account and password should be associated with just one user, unless there's a very good reason for doing otherwise.
- First-time users should receive a unique password (not a default password, and, least of all, an easy guess such as 'password' or 'abc123' or, worst of all, a null password) and be forced to change it at the first login.
- Random patterns should be used for assigned passwords, rather than a 'joe' (same word as the account name) or other easily guessed formula. It's perfectly OK for a password to be unmemorable if the user is obliged to change it at his or her first login.
- Passwords shouldn't be given or changed on the strength of an unverified phone call. Ringing back to a trusted phone number or emailing a trusted individual is better than nothing, but certainly isn't as secure as requiring an individual to report in person with verifiable identification.
- A classic password attack technique is to take advantage of an accessible password database like /etc/passwd and play with the password field to apply guessing techniques off-line. Shadow or dummy password files and similar deceptive techniques are recommended.

Password aging is a good idea in principle. However, it pressures the user into evasive strategies such as:

- Recycling passwords on systems that allow it (sometimes it's just a matter of changing the password a given number of times until it accepts the one that just timed out). A common variation is to recycle with minor and predictable variations (mypassword1, mypassword2, mypassword3...)
- Using the same password on a number of systems and changing them all at the same time. This is subject to the same objection that is often made to a single sign-on. Breaking one password gives an intruder everything.
- Writing the password down and leaving it somewhere accessible and therefore insecure .

It's sometimes useful to consider whether frequent changes are really necessary or desirable. After all, if you're encouraging the use of good password selection and resistance to social engineering attacks, and making it difficult for an attacker to use unlimited login attempts, a good password should remain a safe password for quite a while.

References

1. Craig Hunt. "TCP/IP Network Administration" (Sebastopol, O'Reilly & Associates, 1992). Excerpted at http://www.kcsd.k12.pa.us/technology/password.html
2. David Harley & Andrew Lee. "A Pretty Kettle of Phish" (San Diego, ESET, 2007). http://www.eset.com/download/whitepapers/Pretty_Kettle_of_Phish.pdf; David Harley & Andrew Lee. "Phish Phodder: Is User Education Helping or Hindering?" in "Virus Bulletin Conference Proceedings 2007." (Abingdon, Virus Bulletin, 2007.) http://www.eset.com/download/whitepapers/Phish_Phodder.pdf
3. Simson Garfinkel & Gene Spafford. "Practical Unix & Internet Security 2nd Edition" (Sebastopol, O'Reilly & Associates, 1996)
4. David Harley. "Choosing Your Password" (Haslemere, Small Blue-Green World, 2009.) http://www.securingourecity.org/resources/pdf/choosing_your_password.pdf

ESET Resources

ESET Threatblog (TinyURL with preview enabled):
http://preview.tinyurl.com/esetblog

ESET Threatblog notifications on Twitter:
http://twitter.com/esetresearch

ESET White Papers Page:
http://www.eset.com/download/whitepapers.php