Best Practices for Managing Passwords: Formation

The best password is one that is hard for an attacker to discover but easy for the user to remember. Setting rules that balance these objectives is nontrivial.

Minimum length, prohibited values and complexity rules decrease the likelihood of an attacker successfully guessing a password. The trade-off, however, is that longer, unusual and more-complex passwords are generally harder to remember, increasing calls to the help desk and tempting users to write down their passwords. The enterprise should seek the happy medium that is most appropriate to its security needs and user population.

Minimum Length

A password of eight or more characters is strongly recommended. A password of six or more characters is the minimum acceptable.

Sensitive systems or situations (for example, remote access) demand longer passwords — approximately twice as long.

Prohibited Values

1. A password must not be blank (that is, no password). A possible exception is for software-only user accounts that cannot be used for interactive login — for example, SPECIAL users in z/OS RACF.

2. Default passwords set when a user account is created (for example, equal to the user's default group in z/OS RACF) must be changed immediately. (Similarly, the default password specified for an account, such as those used by an application or database management system, must be changed during or immediately after software installation.)

3. A password must not be the same as any of the following:
   - The user's common name or initials
• Any common name, dictionary word or phrase — this should include names from popular culture: spock, gandalf, laracroft and so on

• Organizational unit ID

• Any common acronym or data string used within the enterprise

• Any of these values reversed, such as drowssap

Note: Some of these values may be implicitly prohibited by complexity rules.

Complexity

1. A password should include at least one lowercase and one uppercase letter. But note that some systems do not respect case, restricting what rules can be used across multiple systems: for example, z/OS RACF passwords are case-insensitive.

2. A password must not include only numbers: These are trivially found by brute-force attacks. (Four-digit personal identification numbers [PINs] look very weak, but remember that a true PIN can be entered only manually — for example, at an automated teller machine or on a one-time password [OTP] token.)

3. A password might include at least one number and at least one punctuation or other special character. But note that only some special characters are valid in a password on some systems, and there might be conflicts between different systems, restricting what rules can be used across multiple systems. For example, z/OS RACF allows only #, @ and $ (or the hex-equivalent local currency symbol). Note also that requiring characters other than letters can make the password less memorable — that is, more likely to be written down — so the net effect of enforcing such a rule might be to reduce security. Finally, requiring a particular kind of character in a particular position is overly restrictive (and reduces the password space, making brute force attacks easier).

4. A password must include no more than two consecutive repeats of the same character. For example, P4ssssss would be invalid (but Mississ3ppi would be valid).

5. Numbers should not be used in place of similarly formed letters (for example, o:0; l/i:1, A:4; s:5) or the corresponding letters on a phone keypad (2:ABC, 3:DEF and so on). This discourages trivial changes to dictionary words (to conform with Rule 1) that an attacker can easily guess or a brute force attack can easily allow for.
6. Datelike formats should be prohibited.

7. Passwords with more than two characters in the same position as in the previous password must be disallowed. This prevents a user from using a simple sequence such as: apple1a, apple2b, apple3c and so on.

Password Audits

It may be desirable — particularly if all password formation rules cannot be enforced on all systems during password change — to periodically check users' passwords for weaknesses using automated tools. These tools typically use dictionaries of multiple languages and file(s) of restricted names, words, phrases, acronyms and data strings. The use of such tools can, however, be contentious. (In May/June 2000, one security vendor asked the Computer Emergency Response Team [CERT] to raise a security advisory about such a tool that an associate of a rival firm was offering.) If an enterprise does use such a tool, it must stringently restrict its use.

Good Password Guidelines

Far more effective than just prohibitive password policy rules are guidelines that help the enterprise's users choose a strong — and memorable — password.

One sound approach is to use the first letter of each word of a favorite song, poem, quotation, etc. to create a good password. Examples:

- We'll be looking flashy in my Mercedes Benz — WblfimMB
- She walks in beauty like the night — Swibln
- In a hole in the ground there lived a hobbit — Iahitgtlah

If required to conform with formation rules, numbers and/or special characters can be substituted or added to the password. Examples:

- Yesterday, all my troubles seemed so far away — Yamtss4a (4 ~ far)
- My name is Ozymandias, king of kings — Mn1Oe1/e1s ("e1" is the starting position of the white king in chess)
- The sky was the color of a television tuned to a dead channel — %s@%c(a%%%adc (letters in the QWERTY row changed to the shift characters above — but this may be problematic if the user moves between, say, PC and Macintosh keyboards, or between different language/region keyboards)
As mentioned above, requiring users to use numbers and/or special characters may cause problems, so it's essential that users find their own ways of deciding how to substitute or add them.

**Random Password Generators**

Some enterprises use programs that randomly generate password values that are intended to be proof against guessing attacks — to provide users with "ready-made" good passwords. These are useful for creating first-use-only passwords — for example, with interactive voice response systems for self-service password reset. However, we do not recommend broader use of these programs.

Some generate passwords that are difficult to remember — for example, zpr38qkv — so that users are inclined to violate the password-handling rule against writing them down. In fact, users likely can remember random passwords, but the problem is that a user looks at zpr38qkv and says, "Oh, that's too hard to remember," and doesn't even try.

Others will generate passwords that are easier to remember, such as by using pronounceable consonant-vowel-consonant sequences: bit59kep. These are, however, still not as memorable as those that users choose themselves and, because they are "chosen" from an artificially reduced range of possible values, are actually easier to guess.

If a random password generator is used, access to the program and its source code must be stringently restricted.

**Key Facts:**

- Enterprises' password management policies should include password formation rules that can be implemented consistently across all systems.

- Long, unusual and more-complex passwords are generally stronger, being more difficult for an attacker to discover by some technical and many nontechnical means.

- However, long, unusual and more-complex passwords are also more difficult for users to remember. Enterprises should, therefore, provide guidelines to make it easier for users to construct strong yet memorable passwords.

- Password formation rules are only one aspect of an effective password management policy.

**Bottom Line:** The enterprise should enforce password formation rules to ensure that passwords are sufficiently complex to deter
guessing by an attacker. These must be complemented by guidelines that help users choose passwords that are both conformant and memorable.