Strong Passwords

Introduction

One of the weakest links in MIT's computer security efforts is your password. Despite the Institute's efforts to keep hackers out of your personal files and away from MIT-only resources (e.g., email, shared files, sensitive data, licensed software), easily-guessed passwords are still a big problem.

As announced in a recent letter to Academic Council and through other channels, MIT now implements a stricter password checking system for Kerberos IDs to combat this problem. Each time you open a new account or change (reset) your password, the system will prevent you from setting a password that is easily cracked.

In addition, MIT now recommends "pass phrases" instead of passwords. Pass phrases are longer, but easier to remember than complex passwords, and if well-chosen can provide better protection against hackers.

Password rules

Your password or pass phrase must conform to the following rules:

- It must be different from your current password.
- It must be 8 characters or longer.
- It must not be based on your Kerberos username.
- It must not be a word that appears in the dictionary.
- It must contain characters from at least two different character classes (upper- and lower-case letters, letters and symbols, letters and numbers, etc.)
- It must be composed of characters in the Roman alphabet or symbols on the US keyboard.
- It must pass a complexity check based on a password dictionary, a technique commonly used by attackers to crack simple passwords.

Creating a pass phrase

A pass phrase is basically just a sentence, including spaces, that you employ instead of a single pass "word." Pass phrases should be at least 15 to 25 characters in length (spaces count as characters), but no less. Longer is better because, though pass phrases look simple, the increased length provides so many possible permutations that a standard password-cracking program will not be effective. It is always a good thing to disguise that simplicity by throwing in elements of weirdness, nonsense, or randomness. Here, for example, are a couple pass phrase candidates:

```
pizza with crispy spaniels
mangled persimmon therapy
```

Punctuate and capitalize your phrase:

```
Pizza with crispy Spaniels!
mangled Persimmon Therapy?
```

Toss in a few numbers or symbols from the top row of the keyboard, plus some deliberately misspelled words, and you'll create an almost unguessable key to your account:

```
Pizza w/ 6 krispy Spaniels!
mang13d Persimmon Th3rapy?
```

Pass phrase hints:

If your pass phrase is based on a well-known slogan, expression, song lyric, or quotation, be sure to customize it with misspellings, bad grammar, invented words, deliberate typos, or oddly placed keyboard symbols. You can learn more ways to mix up words using the tactics outlined in the Creating better passwords section, below.

Your pass phrase should never contain information that would identify you personally, such as Social Security numbers, telephone numbers,
credit card numbers, birth dates, or your Kerberos username. Instead, rely on a phrase that has enough meaning to you that you’ll remember it easily—then mix it up.

Try to avoid phrases composed of common, smaller words. For example, “My dog has long toes,” though long enough to be a decent pass phrase, contains so many small words that a password cracking program might have a better chance of deciphering it. However, “Provincetown is crowded in August!” or “Revere Beach parking is full!” are both acceptable, and easy to remember.

*Note:* Do not adopt any of the sample pass phrases shown above as your own Kerberos pass phrase. They are, for obvious reasons, no longer secure choices for pass phrases.

Creating effective strong passwords

- **Longer passwords are better passwords.** The more characters a password cracking program has to crunch, the harder it is to guess.
  
  *Example:* 1ictsrgry ("All cats are gray")

- **Use an acronym:** choose the first or second letter of your favorite quotation.
  
  *Example:* itsotfdtd ("It's the size of the fight in the dog")

- **Mix letters and non-letters in your passwords.** (Non-letters include numbers and all punctuation characters on the keyboard.)
  
  *Examples:* Idh82po (I'd hate to go), UR1drfj (you are wonderful).

- **Avoid choosing a password that spells a word.** But, if you must, then:
  - Introduce "silent" characters into the word. Example: va7ni9lla
  - Deliberately misspell the word or phrase. Example: choklutt
  - Choose a word that is not composed of smaller words.

- **Add random capitalization to your passwords.** Capitalize any but the first letter.
  
  *Example:* eIeIoH! o.U.Kid

- **A random mix of alphabetical, numeric and symbolic characters.**
  
  *Example:* stiff3open92research12closper

- **An acronym for your favorite saying, or a song you like.**
  
  *Example:* GykOR-66 (Get your kicks on Route 66) or Lliain! (Live! It's Saturday Night!).

- **An easily pronounced nonsense word with some non-letters inside.**
  
  *Example:* slaRoo@Bey or klobinga-dezmin

- **Change your password at least once a year.** Better yet, change your password every few months to shrink your exposure window. You can make three or four passwords if you like, then switch them throughout the year.

- **Don’t use the same password on multiple accounts.** When one site is compromised, hackers try to use those passwords to access accounts on other sites. Don't let one break-in give hackers access to all your accounts.

*Note:* Do not adopt any of the sample passwords shown above (choklutt, va7ni9lla, etc.) as your own Kerberos password. They are, for obvious reasons, no longer secure choices for pass phrases.

Additional strategies for picking passwords that work

The strategies below may help you in picking a password that passes security checks and is still easy to keep track of.

1. **Use a full, non-trivial sentence**
   
   A sentence that is relevant to you, long enough to not be common or simple, and is not a famous quote from a book or movie will usually work.

   *Examples:* I hated my summer vacation at Myrtle Beach with Sophia! Or If I don't finish my dissertation on nano-desserts I will transfer to Stanford. Don't use these examples.

2. **Use a truly random string of letters, numbers, and punctuation**
   
   If you really cannot think of anything that works, you can use a random string of letters, numbers, and punctuation. This will be quite secure but difficult to remember. However, you can use a secure password manager to keep track of it. As a last resort, you can write it down and keep it in a secure location. Although we used to advise you never write down your password, the vast majority of password attacks are now remote network attacks. If you keep it in a secure location, writing it down is a viable alternative. Make sure you change your password if this location is compromised (for example, if you lose your wallet).

Bad passwords - What the system looks for

In addition to the basic length and character class requirements, the system will also check the password for vulnerabilities to password dictionary attacks and simple patterns that make passwords vulnerable to common techniques used by attackers.

The examples below are passwords that the system will reject because they are methods commonly used on ‘easy-to-guess’ passwords.

**Dictionary words**
The password-checking system screens all passwords against its own large dictionary of 2.7 million English and non-English common words. The words in most major languages are represented, spelled forward and backward. This dictionary is regularly updated to cover all words peculiar to the MIT community (such as “MITx”). Any words found in this dictionary are rejected as passwords.

**Random suffixes and prefixes**

Many people attempt to disguise a dictionary word by adding random characters at the beginning or end of the word. The system automatically screens for this technique. For example, the passwords below would not be allowed:

- massachusettsXX
- massachusettsX
- Xmassachusetts
- XXmassachusetts
- XXmassachusettsXX

(No matter what X or XX is.)

**Non-letters as letters**

Many people try to use certain non-letters as letters within their passwords. The system automatically translates all of the following non-letters into letters before looking up words in its dictionary:

- $ = s
- 4 = h
- 2 = a
- 3 = e
- 0 = o
- 1 = l
- 1 = i

Passwords like Ma$$achusetts would therefore be rejected.

**Capitalization**

Kerberos passwords are case-sensitive: uppercase and lowercase letters are considered to be separate letters. Capitalizing random letters in a dictionary word (caRpoRTS) will not, however, fool the screening program. The point is to capitalize letters in a non-word password, in order to provide another layer of complexity against other password-cracking programs.

**Obvious tricks**

The system automatically screens out passwords set in the following manner:

- Passwords based on a dictionary word spelled backward (sttesuhcassam).
- Passwords based on two dictionary words in a row (dogdog).
- Passwords based on the person’s Kerberos username.
- Passwords that are all white space.
- Passwords that contain control characters.
- Passwords that are all numbers.
- Passwords followed and/or preceded by 1 or 2 characters (9cheval, cheval19, 99cheval, cheval199, 99cheval199 etc.)
- Passwords with several repeating characters (aaaaaaa or aaaaabbb or abababab).
- Passwords that do not have more than 4 characters that differ from the previous character by one (1234abcd).
- Passwords with license plate patterns (daaaddd).
- Passwords with social security patterns (dddsdddd).
- Passwords with phone number patterns (dddsdddd or dddsddddddddd).

**Is all this really necessary?**

MIT’s network is under constant and heavy attack from automated password crackers running against MIT’s authentication systems. While MITnet border protection can be effective against many other kinds of security and network attacks, there is no reliable way to protect against outside attempts to guess user passwords. Access from outside networks is vital to the MIT community, and limiting the number of attempts to guess at a user password can quickly result in wide-spread account lockouts affecting legitimate use by members of the MIT Community.

To illustrate the scale of the problem, the MIT network sees approximately 1.5 million attempts daily to guess user passwords, from automated cracking tools using increasingly sophisticated pattern matching and dictionary attack algorithms. As network bandwidth and computing power available to outside attackers increases, so does the frequency and sophistication of these attacks.

**How to change or reset your password**

MIT’s computing infrastructure offers different methods for changing your password. Certain methods may work better for you depending on how long your password is, what kind of computer you use, and other factors, but the Change Your Kerberos Password web application is what MIT recommends and most people prefer.
The Change Your Kerberos Password web page

- Point your browser to [http://mit.edu/password](http://mit.edu/password).
- In the section titled "Change your Kerberos password, using your current password":
  - Enter your Kerberos Username in the box, making sure it’s all lower-case and does not include the @mit.edu you might include in email.
  - Enter your Old Password; remember that passwords are case-sensitive.
  - Enter a new password twice
  - Click the Change your password button

A yellow message bar at the top of the page will inform you if your password change was successful, or if there was a problem with any of the information you entered.

*Note: The bottom section of the above web app allows you to change your password using your (valid) MIT personal web certificate. This can be a convenient alternative if you have forgotten your password, but still have a valid personal web certificate installed on your computer.*

Change your password from Athena

If you use the Athena academic computing environment you can change your Kerberos password from the command prompt using the command `passwd`.

- Open a terminal window and wait until the command prompt appears.
- Enter the command `passwd` and press Enter.
- Follow the prompts to enter your old password and choose a new password.

Change your password from a Mac with Kerberos Extras installed

If you have a Mac OS computer with Kerberos Extras installed you can change your password from the Mac OS Terminal command line or using the Ticket Viewer application.

**Mac OS Kerberos Extras Ticket Viewer**

- Open the Ticket Viewer application on your Mac. If you have Kerberos Extras installed, Ticket Viewer can be found in Applications > Utilities.
- Click on the Change Password button to change your password.

**Mac OS Terminal command line**

- Open a Terminal application window on your Mac. The Terminal application can be found in Applications > Utilities and wait for the Mac OS command prompt to appear.
- Enter the command `kpasswd username` and press the Return key, where `username` is your Kerberos username in all lower case.
- Follow the prompts to enter your old password and choose a new password.

Change your password from a Windows computer with Kerberos for Windows installed

- Locate the MIT Kerberos Ticket Manager application, usually at Start > Programs > Kerberos for Windows > MIT Kerberos Ticket Manager
- Click on the Change Password button to change your password.

Are password managers a good idea?

Yes, as long as you have a strong password protecting all your passwords in your password manager. Most password managers use encryption. If you use a browser-based password manager such as LastPass, you don't have to remember each individual password for your online accounts, but you do need to remember your master password. Be sure to change that master password regularly. Other options for password managers are KeePass and RoboForm. The basic versions of these are free. Consider activating Two-factor Authentication if your password manager supports it.

References

This article borrows heavily (with permission) from Stanford's article SUNet ID Passwords. MIT's new, stricter password requirements are similar to the ones Stanford has had in place for some time now, and they've done a great job documenting requirements and best practices.