Is there a way to password protect individual apps?

Is it possible to password protect a specific Mac application?

For example, I am interested in protecting Mail because even if you cannot retrieve new emails, you still can read all the already received emails.

This is not a question about security. It's about sharing a device within a family. I don't have crucial information. I just want to avoid my girl or son to send accidentally emails from my account or prevent them to read some of them.

osx applications security

edited Dec 11 '11 at 15:12
koiyu
3,177 7 26 60

asked May 12 '11 at 10:11
Rabskatran
2,372 6 26 60

26 This sounds like an example of the XY problem. You want to prevent people from snooping in your email (X), so you're asking about how to protect your email program (Y). What you really might want to ask is `What's the best way to protect my email from snooping?` And the answer to that would be that it's best to completely prevent unauthorized users from using your account -- for example, requiring a password to deactivate the screensaver or after waking the system from sleep. – Austin May 12 '11 at 21:35

3 Upvote for @Austin's comment. Even if you stop Mail.app from opening, you haven't prevented someone from accessing your email. Mail.app keeps your email in plaintext on disk in `~/Library/Application Support` -- if someone wants your email, they'll know to just grab the files from there and make off with those. – Ian C. ♦ May 12 '11 at 22:37

3 @Everybody : This is not a question about security. It's about sharing a device within a family. I don't have crucial information. I just want to avoid my girl or son to send accidentally emails from my account or prevent them to read some of them. – Rabskatran Sep 22 '11 at 14:20

1 That makes it so much clearer - just reading the bare words, this took on a lot of ramifications you simply don't have with children and accidents. I'm curious if in this case, you just set them up their own accounts or used my suggestion to parental control some of the apps on your account... (Or just answer this yourself - you can say what you chose, and people won't likely keep trying to help make answers better assuming you were still looking for a solution) – bmike ♦ Sep 22 '11 at 15:47

4 → Rabskatran: your problem is a basic security problem. Your need is a basic one: control who read your files (which includes your stored E-mail). If I rightly guessed your problem ☂, I'd vote for Mike's advice: **one account for everyone**! It's free :). – daniel Azuelos Apr 1 '13 at 14:30

10 Answers

Regarding your comment on Paul's answer, wanting to leave your computer for a moment: you have to lock your computer. Period.

Open System Preferences, click Security (top row, second to last option), under the "General" tab, check the box for "Require password [immediately] after sleep or screen saver begins".

Then, when you go to walk away from your computer;

Ctrl ^ + Shift ⇧ + Eject ⏏

(Additional notes: Click the above line.)

Lock your Mac. Walk away. Come back, enter your password to unlock it. Secured console.

answered May 12 '11 at 22:04
Jason Salaz
13.6k 10 60 111

3 +1 Paul was the first to mention this in a comment below his own answer, but since he hasn't integrated it into his own answer yet, I'll vote this up. I wish I could give this +3 for `you have to lock your computer. Period.` – Austin May 13 '11 at 4:18

You can just set parental controls on the account and then determine which apps can be used.

Alternatively you could set a password on your screen saver and also define a suitable "hot

corner" to activate the screen saver - that way when you step away from your computer you can just move the mouse into the hot corner to effectively lock the screen such that it require a password to gain access.

edited May 13 '11 at 7:01

answered May 12 '11 at 10:13

Paul R
486 3 11

No, I want to be able to leave my computer logged for a few moment and been sure that nobody can read my received emails (if Mail is not already started) for instance. But more generally, that nobody would be able to launch an App, make some stuff and quit during my absence. – Rabskatran May 12 '11 at 10:15

8 In that case just set a "hot corner" in the screen saver and require a password to exit the screen saver. When you leave your Mac then push the mouse into the hot corner to lock the screen. – Paul R May 12 '11 at 10:17

or set the timeout for the screen saver to be short - Mail will not necessarily be the only app you want to hide – Mark May 12 '11 at 12:10

So it's not possible to set a passoword on an App... – Rabskatran May 12 '11 at 13:03

1 @Paul +1 I would recommend editing that great suggestion about setting up the password protected screensaver directly into your answer. – Austin May 12 '11 at 21:37

It's possible using scripts.

First, you should enable the script menu in the OS X menu bar. Read the "Script Menu" section here: Enable the Script Menu

Now open your Library/Scripts folder and create a file called "run_with_password.rb" with these contents (change "johndoe" to your username):

```
#!/usr/bin/env ruby
# run an app at lower privilege

require 'etc'
require 'find'

# Note: anyone with sudo access will be able to run as this user. But they could do that
anyway.
# run 'id' at the terminal to find out what your username is.
RUN_USER = 'johndoe'

def get_root_info
  root_entry = Etc.getpwnam('root')
  return root_entry.uid, root_entry.gid
end

ROOT_UID, ROOT_GID = get_root_info

def ensure_root
  Process.uid = ROOT_UID
  Process.gid = ROOT_GID
end

def print_user_info
  [
   [:uid, Process.uid],
   [:gid, Process.gid],
   [:euid, Process.euid],
   [:egid, Process.egid],
  ].each do |arr|
    $stderr.puts arr.inspect
  end
end

def set_effective(euid, egid)
  $stderr.puts "setting effective to #{[euid, egid].inspect}"  if $DEBUG
  # must set group first
  Process.egid = egid
  Process.euid = euid
end

def do_privileged(&block)
  orig_euid = Process.euid
  orig_egid = Process.egid
  begin
    $stderr.puts "raising privileges"  if $DEBUG
    set_effective(ROOT_UID, ROOT_GID)
    yield orig_euid, orig_egid
  ensure
    $stderr.puts "lowering privileges"  if $DEBUG
    set_effective(orig_euid, orig_egid)
  end
end

# must be called after ROOT_UID, ROOT_GID are set
def chmod_files_in_dir(mode, dir)
  mode_str = nil
  case mode
  when Integer
```

```
    mode_str = '%o' % mode
  when String
    mode_str = mode
  else
    raise TypeError
  end
  chmod_proc = proc do
    Find.find(dir) {|entry|
      if File.directory?(entry) and entry != dir
        Find.prune  # don't recurse into subdirs
      elsif File.file?(entry)
        $stderr.puts "chmod #{mode_str} #{entry}"  if $DEBUG
        system 'chmod', mode_str, entry
      end
    }
  end
  # assume that if dir is owned by root, the executables are also.
  if File.stat(dir).uid == ROOT_UID
    do_privileged(&chmod_proc)
  else
    chmod_proc.call
  end
end

def main(argv)
  # Important: this is to abort if we're not running as root.
  ensure_root

  app_path = argv.shift or raise "Need path to .app file, e.g. /Applications/Mail.app"
  app_macos_dir = File.join(app_path, 'Contents/MacOS')
  File.directory?(app_path) or raise "#{app_path} is not an app bundle"
  File.directory?(app_macos_dir) or raise "#{app_path} bundle doesn't have expected MacOS
structure"

  pw_entry = Etc.getpwnam(RUN_USER)
  run_uid = pw_entry.uid
  run_gid = pw_entry.gid


  if $DEBUG
    $stderr.puts [:run_uid, run_uid].inspect
    $stderr.puts [:run_gid, run_gid].inspect
    print_user_info
  end

  # Effectively become RUN_USER
  set_effective(run_uid, run_gid)

  if $DEBUG
    print_user_info
  end

  begin
    chmod_files_in_dir('+x', app_macos_dir)
    # 'open' is asynchronous, so the ensure will run immediately after, and before the app
exits.
    $stderr.puts "Running app: #{app_path}"  if $DEBUG
    system 'open', app_path
  ensure
    chmod_files_in_dir('-x', app_macos_dir)
  end
end

if __FILE__ == $0
  $DEBUG = false
  main(ARGV)
end
```

Next, start Script Editor and paste in this code (again changing johndoe to your username):

```
do shell script "ruby /Users/johndoe/Library/Scripts/run_with_password.rb
/Applications/Mail.app" with administrator privileges
```

Save the file into Library/Scripts as "mail_with_password", making sure the File Format is "Script".

Now "mail_with_password" will appear in your script menu. Every time you run it, it will ask you for your password (just like some installers do). After it's done running, it will disable access to the regular Mail application. So run the script once, then try running the Mail app. It won't run. Note that it means ALL users on your machine will be prevented from running Mail directly, not just your user.

If you ever want to allow Mail to run normally again, run this command at the Terminal:

```
sudo chmod +x /Applications/Mail.app/Contents/MacOS/Mail
```

You might be able to omit the "sudo". Use sudo if you get "Operation not permitted". Note that sudo will ask you for your password to allow privileged operation.

Caveats

1. If you didn't need the "sudo" command above to do the chmod, that means a savvy user might be able to figure out how to enable the Mail app again. You can tighten up security by changing the owner of the MacOS/Mail file to root. That's left as an exercise for the

reader.

2. If someone is able to copy the Mail app to your computer (e.g. via USB drive) they can still get access to your mail.
3. The ruby script is meant to work for most OS X application bundles. I don't recommended tweaking the ruby script unless you really know what you're doing because it's doing certain things as root (the privileged user). Tweaking the applescript code should be harmless; but you should know how to adjust the chmod command to make your app directly runnable again.
4. If the path to the app in the applescript file has spaces or other special characters, you'll have to do something like putting single quotes around the whole path.
5. Edit: User Austin suggested that this procedure doesn't protect the .emlx files. I actually don't use the Mail app so I'm not familiar with the data storage. Similar issues apply to all apps - because this solution does not hide user data.

Paranoia

If someone who knows ruby get access to your logged in user, they could modify the ruby script in a way that wreaks all sorts of havoc when you run the script, since it runs as root for part of the time. If you think this might happen, you should make the script only writable by root. You'll also have to make sure someone doesn't replace the script with their own - they can do this if the folder is writable by you. If you're starting to get scared by these warnings and don't know how to protect yourself, you probably should forget about this solution and just remember to lock your screen when you leave the computer.

edited May 13 '11 at 3:04

answered May 12 '11 at 17:16

Kelvin

382 4 14

Wow, 3 upvotes in 4 hours? I didn't expect too many people to use such a hack, especially since you have to create one applescript for each app you want to protect. I'd like to hear how people are using my solution, even if just for learning new tricks. – Kelvin May 12 '11 at 21:37

6 -1, but I hope this is not taken personally. I really like the level of detail here, and it's clear a lot of thought went into this, but I think this solution gives a dangerous false sense of security. One huge caveat that you didn't mention is that it does not prevent access to the plain-text `.emlx` mail files stored in `~/Library/Mail` . – Austin May 13 '11 at 0:08

1 @Austin you're entitled to your downvote of course, but I don't understand your reason. I think my caveats and paranoia sections should be enough to indicate that this is not meant to be a secure solution - it's only a deterrent. I added your caveat, but it's essentially the same idea behind caveat #2. Also, the OP said that he/she didn't care about another person reading "already retrieved emails". – Kelvin May 13 '11 at 3:13

1 I stand in awe of many complicated things I'll never use in real life. This is one - well done Kelvin - you've earned a +1 from me on this. :-) I doubt it will end up the "best" answer possible for this question, but I hope it scores honorable mention. – bmike ♦ May 13 '11 at 3:30

I think you're misunderstanding that part of the question (granted, it could probably be worded better). @Rabskatran -- please correct me if I am wrong, but I am fairly confident that the OP was trying to explain the reason for wanting to password protect the app. To paraphrase, even though a snoop wouldn't be able to retrieve *new* mail (presumably because the account password wasn't stored in Mail), they *would* be able to read *existing* downloaded mail, and that is why an OP wanted the password protection. – Austin May 13 '11 at 3:31

Yes - several ways to password protect your mail are practical. Since you are concerned about children/family members, the easiest might be to just restrict those apps using Parental Controls on *your* account. At some point they can have their own accounts, and you can lock your whole account.

Here are the options I see as workable for the general case of locking apps or the data apps can access.

1. Make your account a protected parental account and white list the apps you want to allow. You will know the separate admin user/password to allow launch of prohibited apps. **Voila - any app you want is now password protected.**
2. Move the app into a password protected disk image and then make an alias to store in the Applications folder. (deleting the original app first) When any program tries to access the app, you get a chance to enter a password and finder will mount the disk image. You can also script permission changes and other technical trickery to require a password before running the script to make the app runnable again.
3. Store the application data in a password protected encrypted disk image. Here are some common apps and the folders where they store user data.
4. Store your mail app on a removable drive - there is a cottage industry about making standalone app packages to run apps from USB drives.

Keep in mind things like spotlight and other apps using frameworks within apps won't work so well until the images are mounted. If your user password is secure (from the people you

don't want seeing the data) then you can store the disk image passwords in the keychain.

Also - unless you protect the data files - it's only security by obscurity and someone could copy your data elsewhere or just look at it from spotlight or other apps like text edit. They also could bring a copy of the mail (or whatever other) app from another computer. Apps can run from anywhere and not just the Applications folder once an admin user blesses them for the first run on that system.

Therefore #3 is the only way to go. Lock up your data and don't worry about the apps.

edited Sep 22 '11 at 19:13

answered May 13 '11 at 3:28

bmike ♦

95.7k 34 154 363

Here is a Mac utility that will do what you're asking for. It can password protect individual apps. Also, you can set a timeout value that will exit the application after being inactive for the selected time.

Mac App Blocker

answered Dec 10 '11 at 17:53

Ken

29 1

Mac App Blocker is exactly what he is looking for. As an Apple tech, we've been getting this request for this ability for years (remember Oulook 2001?). Users aren't concerned that their kids or co-workers will go digging into Library folders to pull .emlx files out and read them. That's not the point - they don't mind letting others use their Mac, they just don't want them reading their emails. Plain and simple. The Mac App Blocker does what it does and that's all they need it to do. – user22134 Apr 26 '12 at 21:36

Hmm. I've wanted to do this for a while now. Let's take another example that doesn't keep local files unless it's told to - chat programs like Skype or Trillian, for example. The trouble with parental controls is you can't be an admin user on the machine AND have parental controls on your admin user account.

My suggestion would be to simply do the following:

```
chmod 600 /Applications/Trillian.app ; chown `whoami`:staff /Applications/Trillian.app
```

This stops anyone from running it, but marks it as yours

Then to run it,

```
sudo /Applications/Trillian.app/Contents/MacOS/Trillian
```

This gets past the 600 (rw- --- ---) access permissions above, and you will have to enter an admin password to run it.

You might also look into the setuid bit and see if that helps (make the app run as another user, in other words).

edited Sep 22 '11 at 19:00

Jason Salaz

13.6k 10 60 111

answered Sep 22 '11 at 13:39

John

11 1

by the way, the "whoami" above must have grave accents or backticks on either side of it. The anti-hacking stuff on this website stripped those marks. – John Sep 22 '11 at 13:40

1 Fixed. Feel free to hit edit and see how I fixed it. – Jason Salaz Sep 22 '11 at 19:00

What about setting up separate user accounts with their own apple id, email with parental restrictions or no email at all. Apps can be shared by different accounts. Depending on the age of your kids, if they have administrator ability i.e. your account, they might be able to harm your account without meaning to. With you being the administrator you can control what they can do with their account.

answered Jan 17 '13 at 13:52

rclyde

405 3 8

Mac App Blocker

Does exactly what you're asking for. From their website:

With Mac App Blocker, you can password-protect EACH application on your Mac. Keep your apps and your Mac safe. Set a timeout value to automatically exit the protected application so even when you leave your computer unattended, you're still protected.

edited Feb 16 '14 at 16:00

Ian C. ♦
24.9k 13 89 161

answered Feb 16 '14 at 15:50

Sherif
11 1

Answers on Ask Different need to be more than just a link. It's okay to include a link, but please summarize or excerpt it in the answer. The idea is to make the answer stand alone. – patrix ♦ Feb 16 '14 at 15:57

Thanks to @Ian C. for making a sentence of this link fallen from a copy-paste. – daniel Azuelos Oct 4 at 9:26

It's an old thread I know, but I had the same problem like you… Here is the solution: An application called iLock. Protects single apps with a password after your choice. Simple, efficient and completely free! https://www.macupdate.com/app/mac/49881/ilock

answered Apr 16 '14 at 14:14

user75894
11 1

Separate accounts is the solution to this problem.

Email might be the obvious one, but there'll be some important files, or bookmarks, or open banking tabs, etc that could be equally bad for a child to randomly delete/open/mess with.

Separate accounts (with only parents as administrators) means you can keep everyone's files, emails, settings, etc separate, even though applications are accessible by all.

If user X uses the Mail application, their emails and account settings are only accessible when they are logged in. When user Y opens Mail, they will have their own accounts/mail, even though both are using the same Mail application.

You might have a shared account for adults, and another for kids, or individual accounts for each person. With separate accounts, you will want to lock the computer (via screensaver, or returning to the login screen) when you leave it. But even if you forget, your kids will want to use their own account (with their own bookmarks, or save game files, or high scores, or documents, etc) rather than yours, which will help prevent any accidental access/damage. Even something as simple as not having to log out of facebook from whoever last used the browser (each account has its own browser settings, which remember who is logged into Gmail, facebook, icloud, etc).

You can also prevent some users from accessing particular applications, websites, etc via Parental Controls in System Preferences.

answered Jan 8 '14 at 6:15

drfrogsplat
4,554 2 20 43