Practical Paranoia™ macOS 10.13

Marc L. Mintz, MBA-IT, ACTC, ACSP

TPP
The Practical Paranoid

Practical Paranoia: macOS 10.13 Security Essentials

Author: Marc Mintz

Editions: v1.0 20170918 • v1.01 20170923 • v1.1 20171001 • v1.2 20171022 • v1.3 20180325 • v2 20180420

Cover design by Ed Brandt

ISBN-10: 1976513650
ISBN-13: 978-1976513657

Dedication

To Candace,
without whose support and encouragement
this work would not be possible

Contents At A Glance

Contents In Detail

4 Passwords

For a people who are free, and who mean to remain so, a well-organized and armed militia is their best security.

–Thomas Jefferson[1]

Knowledge, and the willingness to act upon it, is our greatest defense.

–Marc L. Mintz[2]

What You Will Learn In This Chapter

- Create a strong password
- Use the Keychain
- View an existing Keychain record
- Challenge questions
- Store challenge Q&A in Keychain
- Access secure data from Keychain
- Harden the Keychain
- Synchronize Keychain across macOS and iOS devices
- Use LastPass to save website credentials
- Create Password Policies

[1] *https://en.wikipedia.org/wiki/Thomas_Jefferson*
[2] *https://mintzit.com/*

4.1 The Great Awakening

In June 2013, documents of NSA origin were leaked to The Guardian newspaper[3]. The documents provided evidence that the NSA was both legally and illegally spying on United States citizens' cell phone, email, and web usage. These documents, while causing gasps of outrage and shock by the public, revealed little that those of us in the IT field already did not know/suspect for decades: every aspect of our digital lives is subject to eavesdropping.

The more cynical amongst us go even further, stating that *everything* we do on our computers *is* recorded and subject to government scrutiny.

But few of us have anything real to fear from our government. Where the real problems with digital data theft come from are local kids hijacking networks, professional cyber-criminals who have fully automated the process of scanning networks for valuable information, competitors/enemies and malware that finds its way into our systems from criminals, foreign governments, and our own government.

The first step to securing our data is to secure our computers and mobile devices. Remember, we are not in Kansas anymore.

[3] *https://en.wikipedia.org/wiki/NSA_warrantless_surveillance_controversy*

4.2 Strong Passwords

We all know we need passwords. Right? But do you know that *every* password can be broken? Start by trying *a*. If that does not work, try *b,* and then *c.* Eventually, the correct string of characters will get you into the system. It is only a matter of time.

Way back in your great-great-great grandfather's day, the only way to break into a personal computer was by manually attempting to guess the password. Given that manual attempts could proceed at approximately 1 attempt per second, an 8-character password became the standard. With a typical character set of 24 (a–z) this created a possibility of 24^8 or over 100 billion possible combinations. The thought that anyone could ever break such a password was ridiculous, so your ancestors became complacent.

This is funny when you consider that research has shown that most passwords can be guessed. These passwords include: name of spouse, name of children, name of pets, home address, phone number, Social Security number, and main character names from Star Trek and Star Wars (would I kid you?). Most computer users are unaware that what they thought was an obscure and impossible-to-break password could be cracked in minutes.

It gets worse. A while back the first hacker wrote password-breaking software. Assuming it may have taken 8 CPU cycles to process a single attack event, on an old computer with a blazing 16 KHz CPU that would equate to 2,000 attempts per second. This meant that a password could be broken in less than 2 years. Yikes.

IT directors took notice.

So down came the edict from the IT Director that we *must* create *obscure* passwords: strings that include upper and lower case, numeric, and symbol characters. But in many cases, this was a step backward. Since a computer user could not remember that their password was 8@dC%Z#2, the user often would manually record the password. That urban legend of leaving a password on a sticky note under the keyboard? I have seen it myself more than a hundred times.

Come forward to the present day. A current quad-core Intel i7 with freely available password-cracking software can make over 10 billion password attempts

per second. Create an army of infected computers called a botnet to do your dirty work[4] and you can likely achieve over a hundred trillion attempts per second, unless your system locks out the user after x number of failed log on attempts.

What does this mean for you? The typical password using upper and lower case, number, and symbol now can be cracked with the right tools in under than 2 minutes. If using just a single computer to do the break in, make that a week. Don't believe it? Look at the *haystack*[5] search space calculator.

If we use longer passwords, we can make it too time consuming to break into our system, so the bad guys will move on to someone else.

But you say it is tough enough to remember *8* characters, impossible to remember more?

This is true, but only if we keep doing things as we have always done before. Since virtually all such attacks are now done by automated software, it is only an issue of length of password, not complexity. So, use a passphrase that is easy to remember, such as, "Rocky has brown eyes" (which at 100 trillion attempts per second could take over 1,000,000,000,000,000 centuries to break – provided Rocky is not the name of your beloved pet and thus more guessable).

How long should you make your password, or rather, passphrase? As of this writing, Apple[6], Google[7] and Microsoft[8] recommends a minimum of 8 characters. US-CERT[9] [10] currently recommends at least 15 for administrative accounts, at least 8 for non-administrators. Cisco recommends[11] at least 8. My recommendation to clients is a minimum of 15, in an easy-to-remember, easy-to-enter phrase.

---

[4] *http://en.wikipedia.org/wiki/Botnet*
[5] *https://www.grc.com/haystack.htm*
[6] *https://support.apple.com/en-us/HT201303*
[7] *https://support.google.com/a/answer/33386?hl=en*
[8] *https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/Microsoft_Password_Guidance-1.pdf*
[9] *https://security.web.cern.ch/security/recommendations/en/passwords.shtml*
[10] *https://www.us-cert.gov/ncas/alerts/TA11-200A*
[11] *http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_aaa/configuration/15-sy/sec-usr-aaa-15-sy-book/sec-aaa-comm-criteria-pwd.html*

In addition to password length, it is critical to use a variety of passwords. In this way, should a bad entity gain access to your Facebook password, that password cannot be used to access your bank account.

Yes, soon you will have a drawer full of passwords for all your different accounts, email, social networks, financial institutions, etc. How to keep all of them organized and easily accessed amongst all your various computers and devices? More on that later in the *LastPass* section of this *Password* topic.

Apple Password Recommendations

- Maintain an 8-character minimum length
- At least one number
- Include both upper and lowercase letters
- For a stronger password, add additional characters and punctuation marks

Microsoft Password Recommendations

- Maintain an 8-character minimum length
- Eliminate character-composition requirements
- Eliminate mandatory periodic password resets for user accounts
- Ban common passwords, to keep the most vulnerable passwords out of your system
- Educate your users not to re-use their password for non-work-related purposes
- Use multi-factor (2-factor) authentication

US-CERT Password Recommendations

- Private and known only by one person
- Not stored in clear text in any file or program, or on paper
- Easily remembered
- At least 15 characters long for administrators, at least 8 characters long for non-administrators

- A mixture of at least 3 of the following: upper case, lower case, digits, and symbols
- Not listed in a dictionary of any major language
- Not guessable by any program in a reasonable time frame

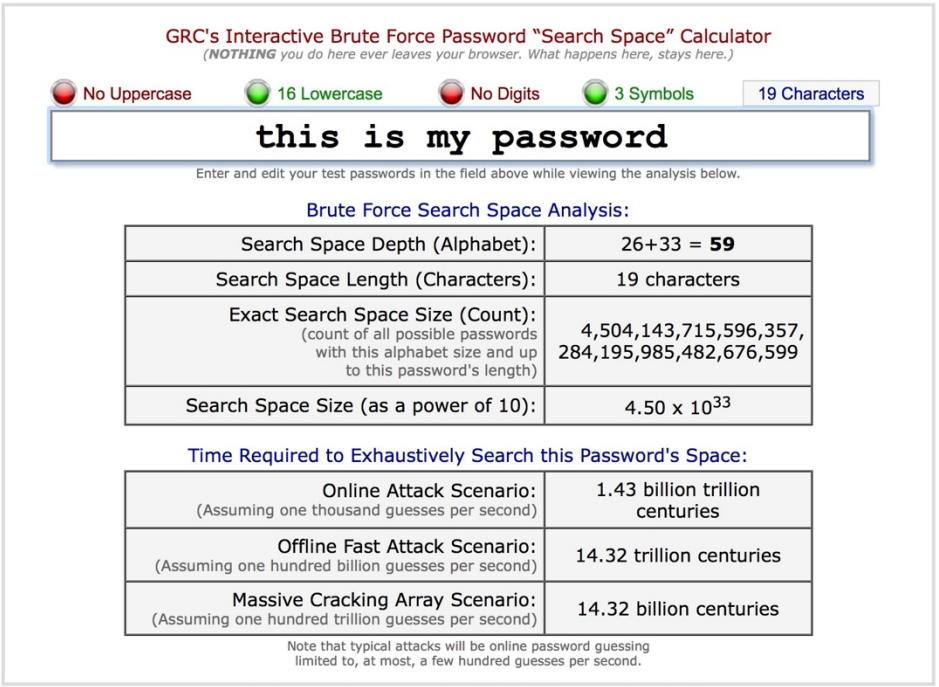
4.2.1 Assignment: Create a Strong User Account Password

As password cracking is now done through automated software, complexity isn't nearly as important as it was when humans were attempting the crack. This is to say that a password of 11111111111111 is about as secure as f^w1&%Ge0*$W18. I recommend using a passphrase–easy to remember, easy to enter, at least 15 characters. For example, *I love brown eyes* is an excellent password.

In this assignment, you create a strong password for your computer account.

1. Think up a password for yourself that is consists of at least 15 easy-to-remember and easy-to-enter characters, and meets the strength/complexity required by your organization.

2. Test how difficult it is to break your password by visiting haystack at *https://www.grc.com/haystack.htm*.



GRC's Interactive Brute Force Password "Search Space" Calculator

(**NOTHING** you do here ever leaves your browser. What happens here, stays here.)

No Uppercase | 16 Lowercase | No Digits | 3 Symbols | 19 Characters

this is my password

Enter and edit your test passwords in the field above while viewing the analysis below.

Brute Force Search Space Analysis:

Search Space Depth (Alphabet):	26+33 = **59**
Search Space Length (Characters):	19 characters
Exact Search Space Size (Count): (count of all possible passwords with this alphabet size and up to this password's length)	4,504,143,715,596,357, 284,195,985,482,676,599
Search Space Size (as a power of 10):	4.50×10^{33}

Time Required to Exhaustively Search this Password's Space:

Online Attack Scenario: (Assuming one thousand guesses per second)	1.43 billion trillion centuries
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	14.32 trillion centuries
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	14.32 billion centuries

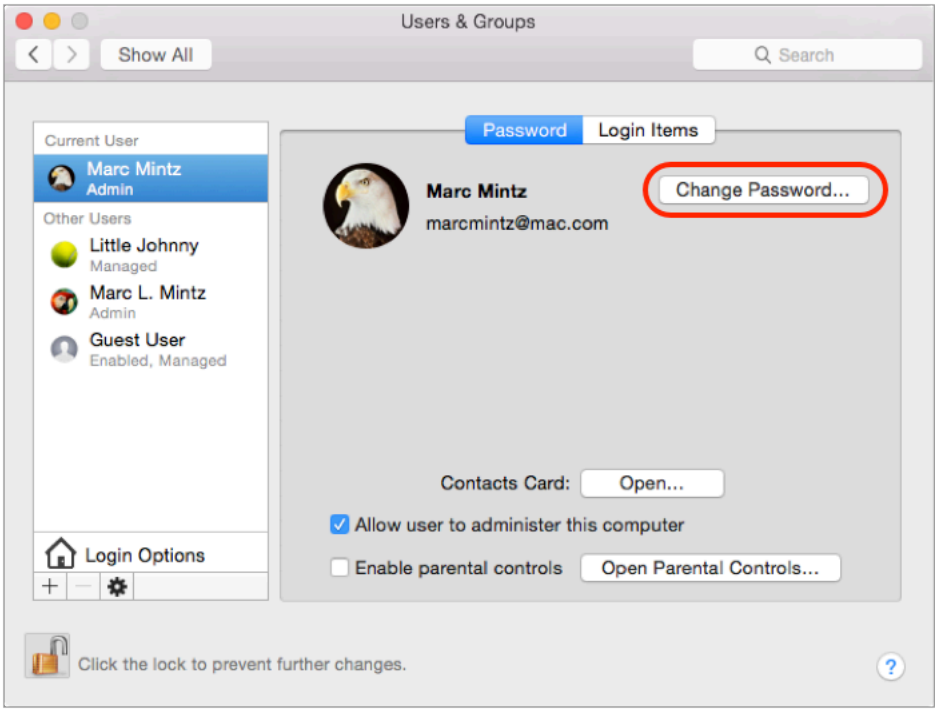
Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.

3. Record your new password in a way that is secure, and you can find when you need it. I recommend using LastPass (more on that later in this chapter), or Apple Contacts.
4. Exit the browser.

Change Your Old Password to the Strong Password

5. Log in to your computer using your user account.
6. Click on *Apple* menu > *System Preferences* > *Users and Groups*.

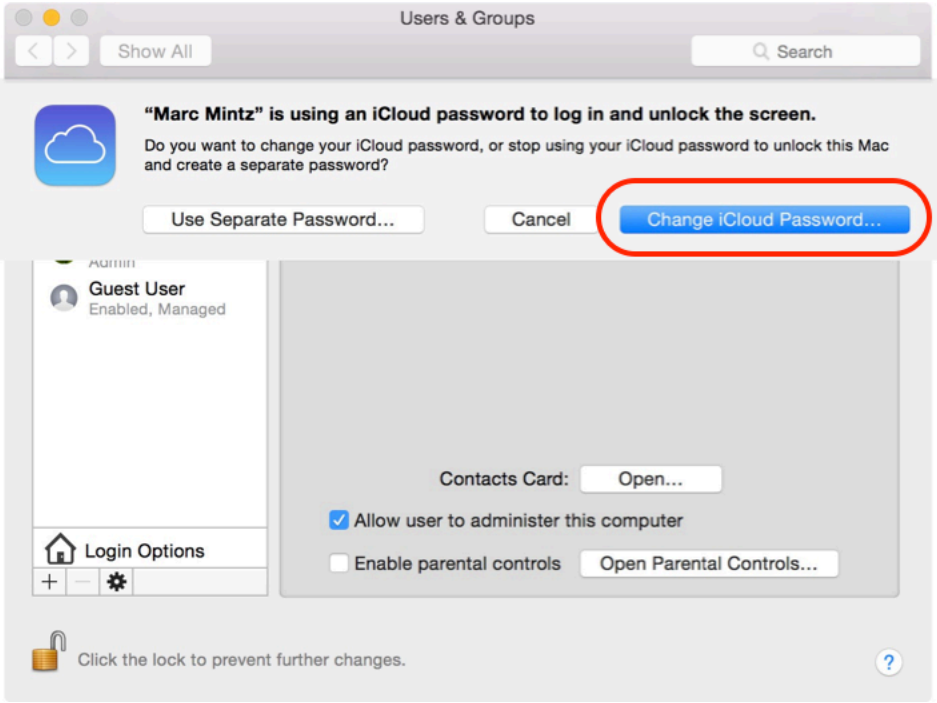
7. Select the Change *Password* button:



- Note: When changing a user/login password, if possible, the change should be made while logged in with that user account. Doing so will simultaneously change the *Keychain* password to match. The Keychain stores usernames and passwords. When changing the user/login password in any other way, the Keychain password remains unchanged. If the user doesn't then know the password to the Keychain, it is impossible to ever open again, and all stored passwords will be lost. More on Keychain later in this chapter.

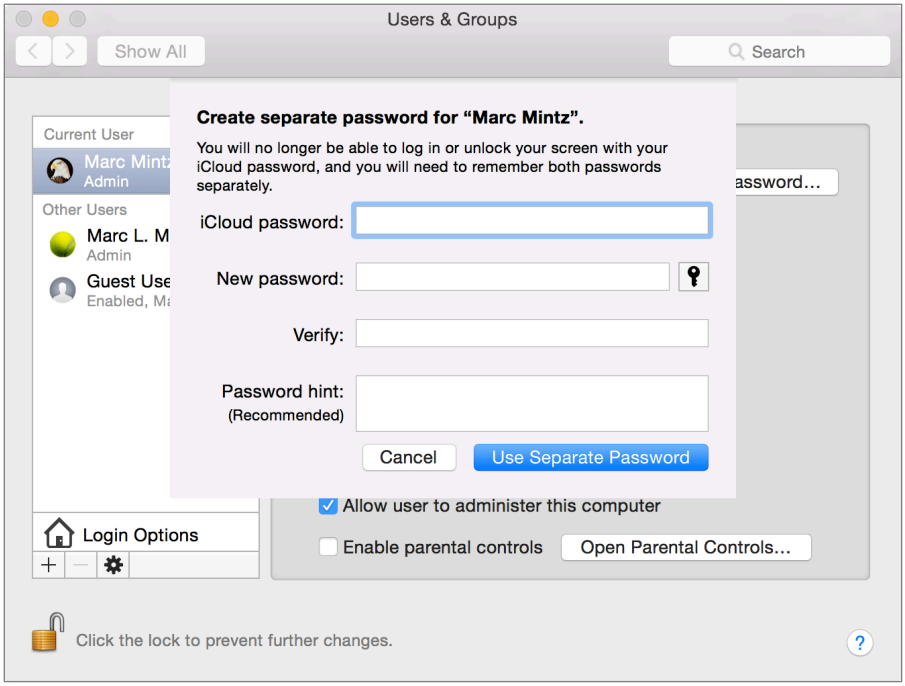
8. By default, your login password is set the same as your iCloud password. You will be asked if you want to *Use Separate Password…*, or to *Change iCloud Password…*

   a. Synchronizing the iCloud and login password makes remembering both easier, and accessing your iCloud data from a new computer easier, but it also presents a roadblock to login should the Apple authentication servers be offline (as has happened at least once).



b. If you select Change iCloud Password, a browser opens to the My Apple ID page at Apple so that you may manage your ID.

c. If you select *Use Separate Password, the Create separate password for "<user name>"* window appears so that you may create a password. At the prompt, enter your *iCloud password, New password, Verify* your new password, and then select the *Use Separate Password* button:



9. Quit System Preferences.

Your new, strong password now is in effect.

4.3 Keychain

In our grandparent's day, life was so much simpler. I'm not talking about politics or sociology, but, well… to give an example: My grandfather had four keys in his pocket at all times: one for home, one for the car, and the other two he could never remember what for.

In today's world, the realm of keys has expanded into the digital world. You now have keys or passwords for logging on to your computer, your phone, your tablet, your email, many of the websites you visit, Wi-Fi access points, servers, your frequent flyer account, etc. In my case, I have 857 passwords in use. I know because they are all neatly stored in a database so that I don't have to remember them.

Unfortunately for most of us, our "keys" are not very well organized, so when we need to access our mail from another computer, or order a book on Amazon, we are stuck.

By default, your Mac stores most usernames and passwords used to access Wi-Fi networks, servers, other computers, and websites. The exceptions are usually websites that are programmed specifically so they do not have credentials saved. These are typically financial institutions.

The built-in tools that store this information automatically can also be used to manually store any text-based data. This includes credit card information, software serial numbers, challenge Q&A, offshore banking information, etc.

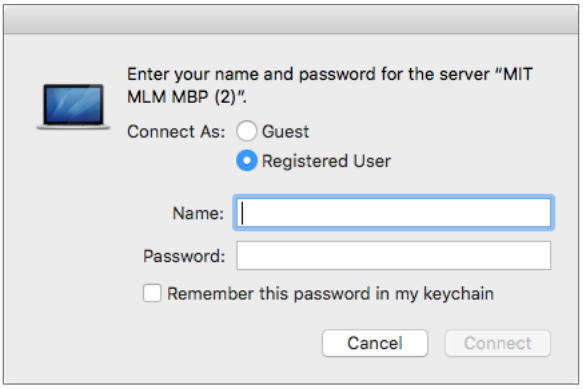
Your Mac has two locations to store keys:

- Safari, which stores only credentials for websites visited with Safari.
- Keychain database, which stores username, password, and URL for websites which request authentication, Wi-Fi networks, servers, other computers you access, email accounts, and encrypted drives.
    - Located at *~/Library/Keychain*
    - Opened with the *Keychain Access* utility

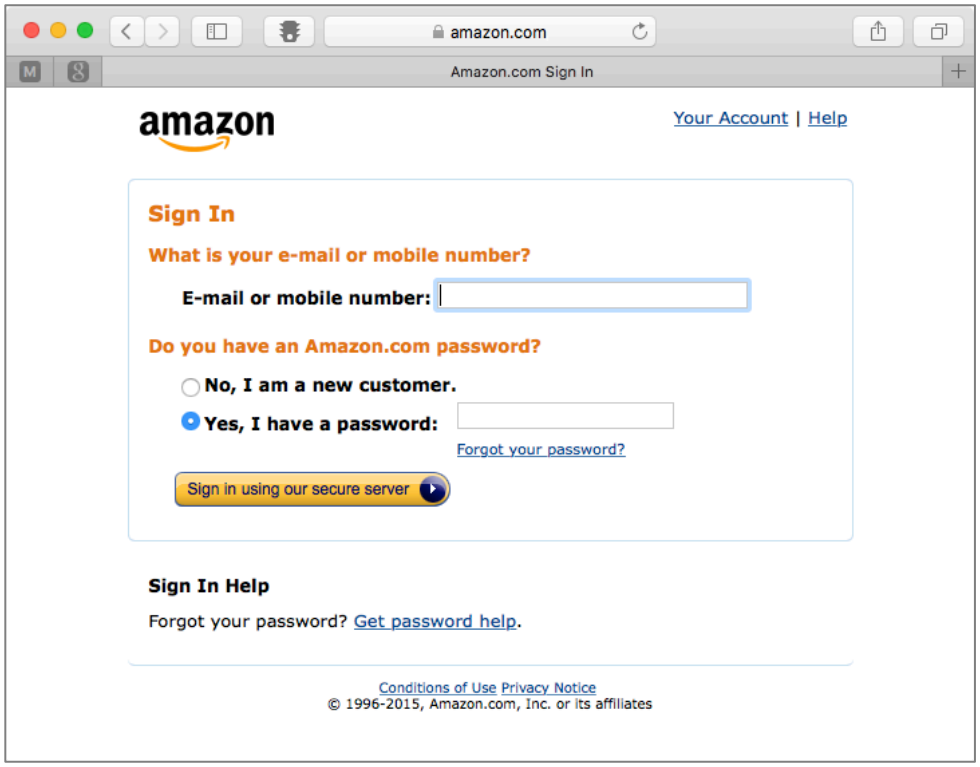
Keychain is what interests us here.

Let's take the case of visiting a website that requires a username and password, connecting to another computer or server, or performing some other action that triggers an authentication request. The following are the steps as they typically occur:

1. A prompt appears requesting a username and password.
    - Typical default authentication window for a server:



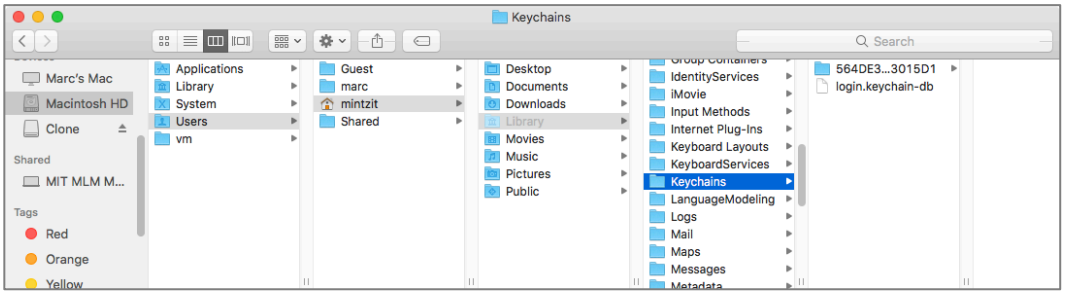
- Typical authentication window for a website:



2. Enter your username and password. In most cases, there is a checkbox to *Remember this password in my Keychain.* Enable that checkbox, and then click Enter or Continue.
3. The website takes you to the appropriate secured page or the other computer mounts a drive on your Mac.

Behind the curtain, your Mac has copied your username and password into the Keychain database, named *Login.Keychain*.

This database is in your Home *Library/Keychains* folder. The database is military grade AES 256 encrypted, safe from prying eyes.



The next time you visit this same website or server, the steps change somewhat:

1. You surf to the website or select a server to access.
2. A prompt appears requesting a username and password.
3. Behind the scenes your web browser or Finder asks: "Has the Keychain stored the credentials for this site or server?"
4. A query is made of the Keychain database based on the URL of the site or the name of the server.
5. If Keychain has stored the username and password associated with the URL or server (it has), the credentials are automatically copied/pasted into the *username* and *password* fields.
6. Select *Enter*.
7. The website takes you to the appropriate secured page or the server share point mounts.

Note that you did not need to know your credentials–Keychain did it all for you.

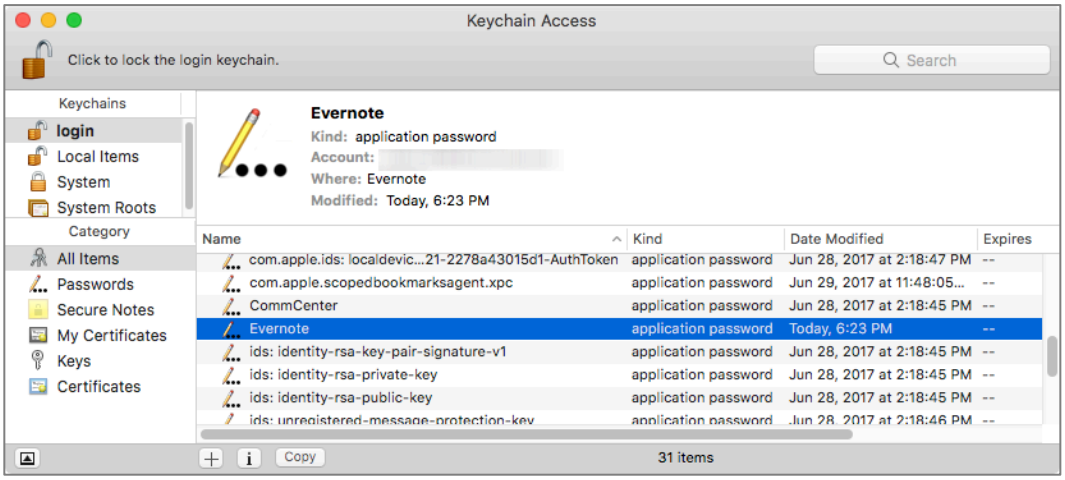
macOS ships with a tool allowing the user full access to the database, named *Keychain Access*, located in the /Applications/Utilities folder.

4.3.1 Assignment: View an Existing Keychain Record

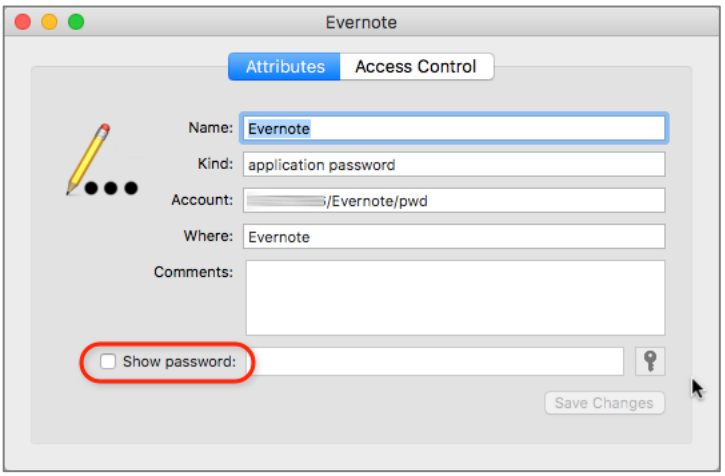
Perhaps a trusted visitor needs access to your Wi-Fi network, and you have forgotten the password to that network. The Keychain database has it stored, you just need to look for it.

In this assignment, you examine a record in the Keychain.

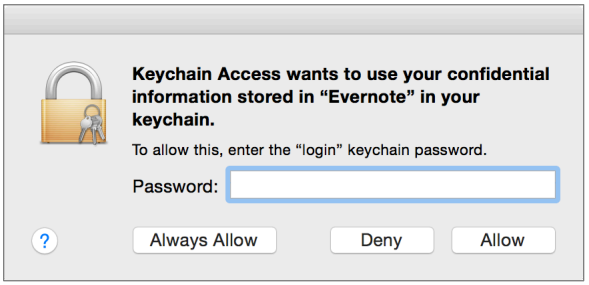
1. Launch *Keychain Access* (located in /Applications/Utilities/).
2. From the sidebar, in the *Keychains* field, select log in. This is the database that holds your secure information.
3. From the sidebar, in the *Category* field, select *All Items.*
4. In the center, main area of the window, double-click on the *target record*, in this example, *Evernote*.



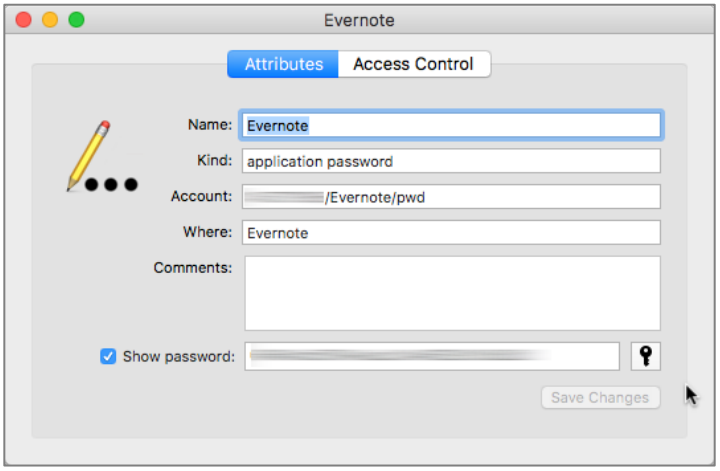
5. The records *Attributes* window will open. At the bottom of the *Attributes* window you will see *Show Password*. Enable the checkbox. This will open the authentication window.



6. At the prompt, enter your Keychain password. By default, this is the same as your user account password. This will authorize Keychain to show you the password. Then click the *Allow* button.



7. The *Show Password* field will now display the needed password.



8. Quit Keychain Access.

4.4 Challenge Questions

A Challenge Question is a way for websites to authenticate who you claim to be when you contact support because of a lost or compromised password.

For example, when registering at a website you may see: *Question – Where did your mother and father meet?*

The problem with this strategy is that most answers easily are discovered with an Internet search of your personal information, or a bit of social engineering.

The solution is to give bogus answers. For example, my answer to the question; *Where did your mother and father meet?* may be; *1954 Plymouth back seat.* It would not be possible for a hacker to discover this answer, as it is completely bogus. My mother tells me it was a 1952 Dodge.

Unless you are some type of savant, there is no way you will remember the answers to your challenge questions. But, there is no need to remember. We already have a built-in utility that is highly secure and designed to hold secrets such as passwords–Keychain Access!

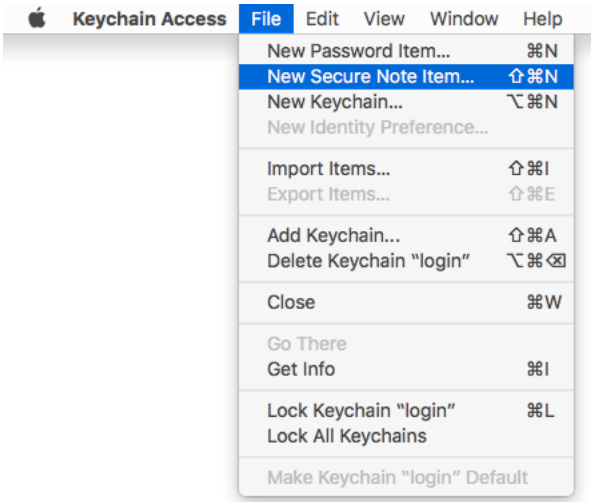
Although Keychain can automatically record and auto fill usernames and passwords, it will require manually entering other data such as challenge Q&A.

4.4.1 Assignment: Store Challenge Q&A in the Keychain

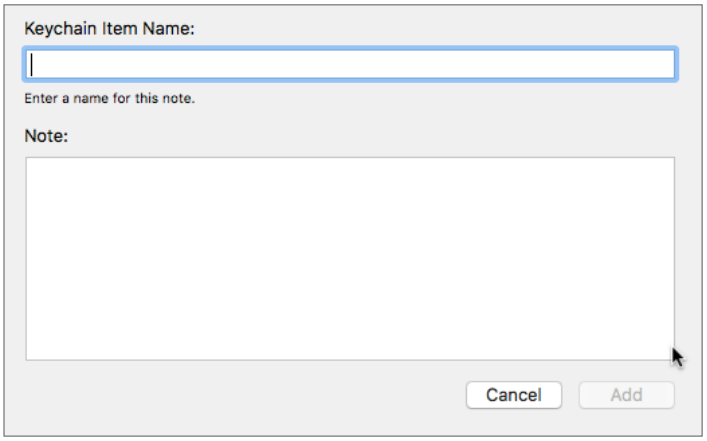
In this assignment, you manually store the challenge Q&A for a pretend website, myteddybear.com.

1. Open *Keychain Access.app*, located in */Applications/Utilities.*

2. Select the Keychain Access *File* menu > *New Secure Note item*…



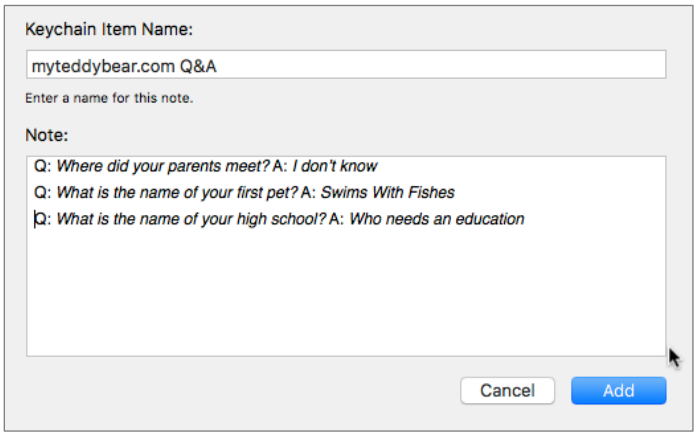
3. The Keychain *Item Name* window appears.



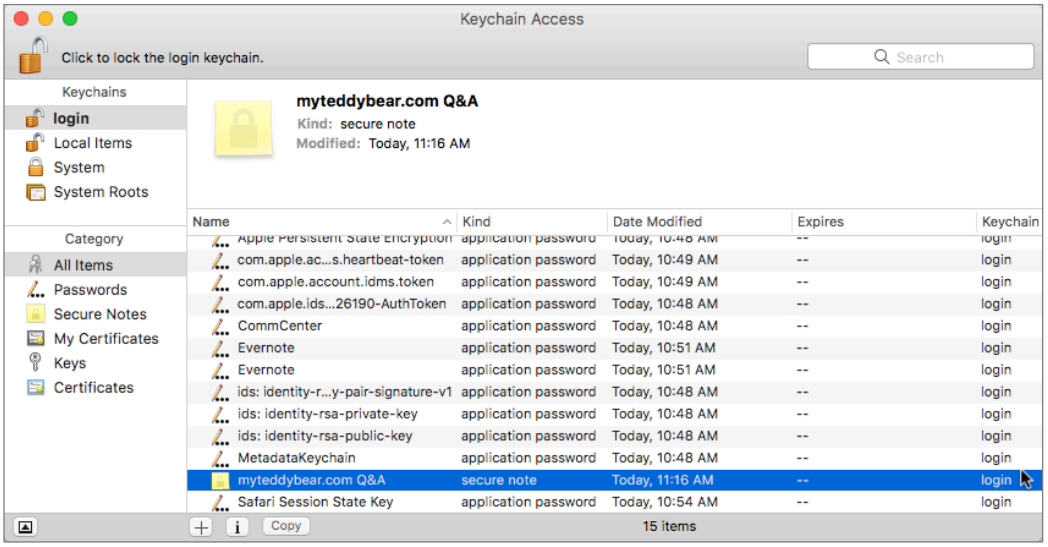
4. In the *Keychain Item Name* field, enter: *myteddybear.com Q&A*.

5. In the *Note* field, enter:
Q: *Where did your parents meet?* A: *I don't know*

Q: *What is the name of your first pet?* A: *Swims With Fishes*
Q: *What is the name of your high school?* A: *Who needs an education*



6. Select the *Add* button.
7. You will find your new Secure Note within all your other Keychain items.



8. Quit Keychain Access.

Your challenge questions and answers are now securely stored.

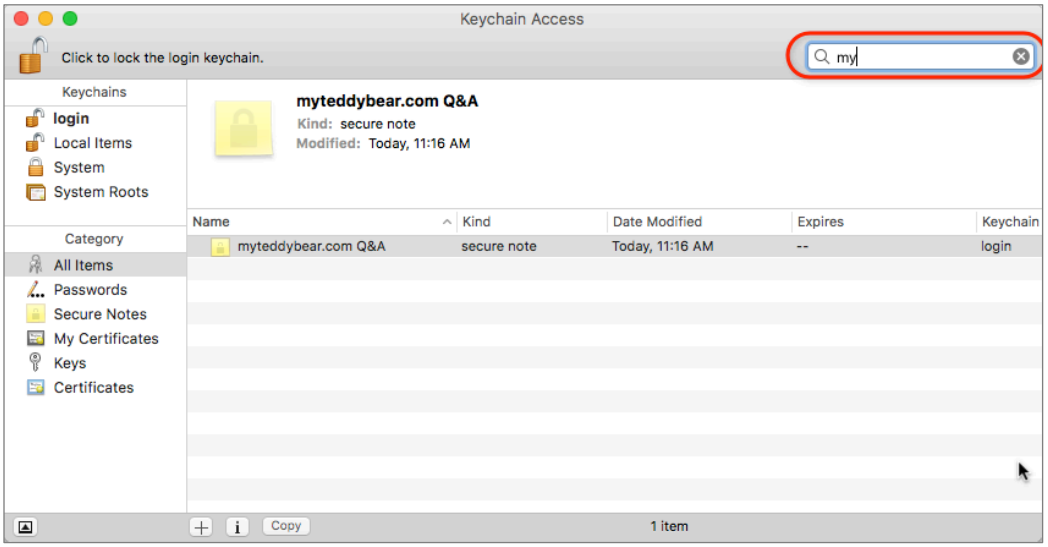
4.4.2 Assignment: Access Secure Data from Keychain

There may come a time that you forget your password to myteddybear.com. A call to technical support with a request to either retrieve or reset your password is met with a challenge question. If you are like me, your synapses holding that memory have long died out.

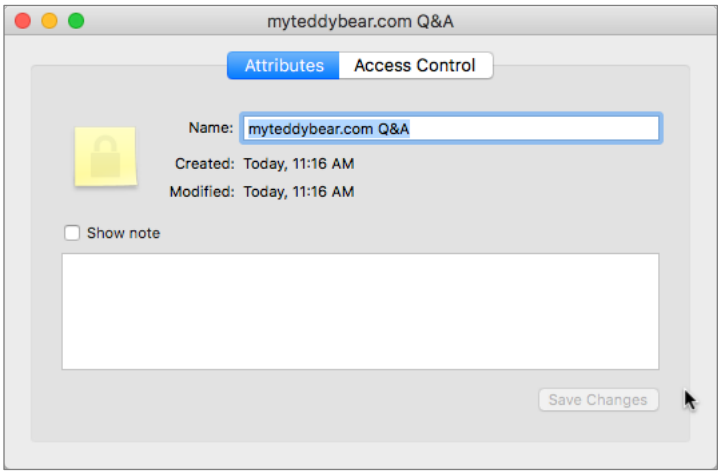
But, no worries! You do remember that you have the habit of storing all your important data securely in your Keychain.

In this assignment, you retrieve your challenge Q&A for myteddybear.com.

1. Open Keychain Access.app, located in */Applications/Utilities*.
2. Click in the *search* field at the top right corner of the *Keychain Access* window.
3. Enter: *myteddybear*. As you type, only those records matching your search string appear, until only the proper record shows.

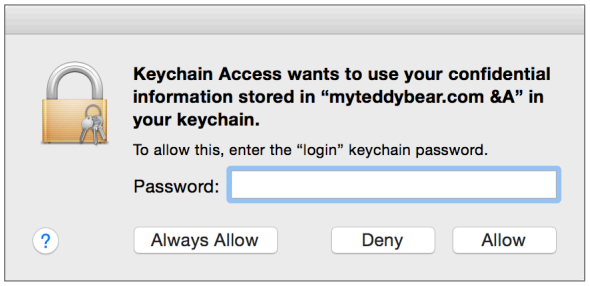


4. Double-click on the myteddybear.com record to open it. Your password is not initially displayed. This is intentional, doubly protecting your data.

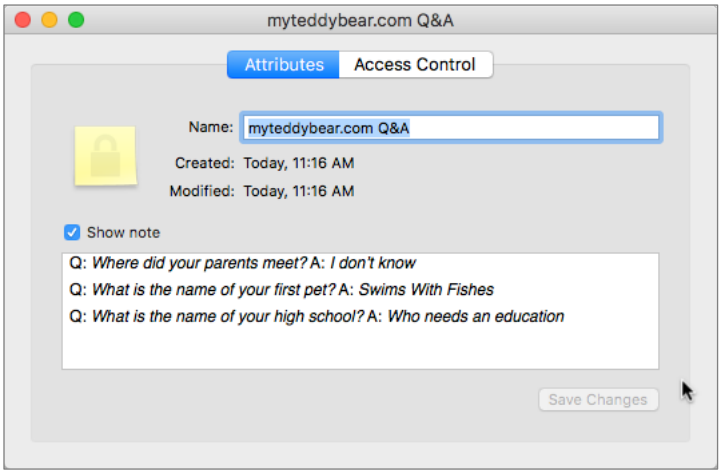


5. Enable the *Show note* checkbox.

6. You are prompted to enter your Keychain password. By default, this is the same as your log in password. Enter your Keychain password, and then click either the *Always Allow*, or *Allow*, button. By selecting *Always Allow*, you will not be asked to verify your Keychain password for this record in the future. If you select *Allow*, you have access to your data, but you will be prompted for your Keychain password in the future.



7. After selecting either *Always Allow* or *Allow,* you see your challenge Q&A.



8. Close the window and Quit *Keychain Access*.

4.5 Harden the Keychain

The work we have done so far in Keychain Access is all that is necessary for almost every environment. Some situations call for even greater levels of security–think military bases, the computer used by the CEO, and my aunt Rose who needs to protect her secret recipe for kosher raisin noodle Koogle.

There is an option to further protect the Keychain–have your Keychain automatically log off after X minutes of inactivity.

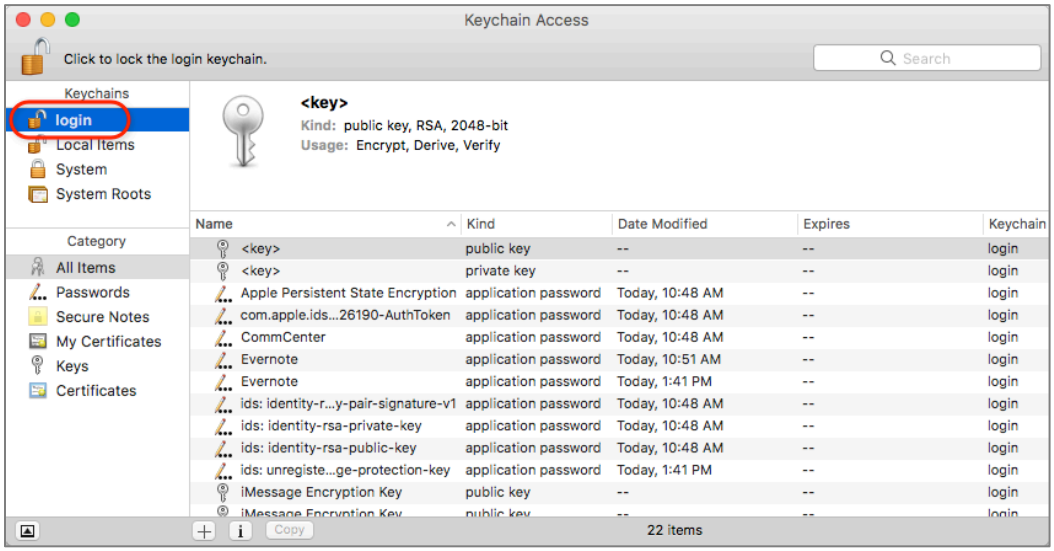
By default, the Keychain remains unlocked if the user remains logged in. There is the option to set the Keychain to automatically lock after a specified amount of inactivity time.

Let's say Keychain Access to automatically lock the Keychain after 5 minutes of inactivity. Upon log in, if the Keychain password is the same as the log in password, the Keychain will unlock and remain unlocked for 5 minutes. If you need an auto fill from data held in Keychain after that 5 minutes, you are prompted for the Keychain password. If within 5 minutes another auto fill is needed, the data is pulled from Keychain automatically. But when 5 minutes or more has passed, the Keychain will lock automatically.

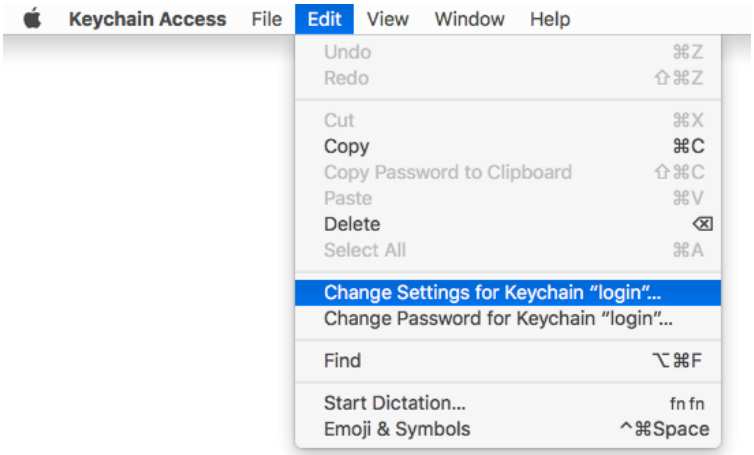
4.5.1 Assignment: Harden the Keychain With a Timed Lock

In this assignment, you give your Keychain a timeout to automatically lock after it has not been used for 1 minute.

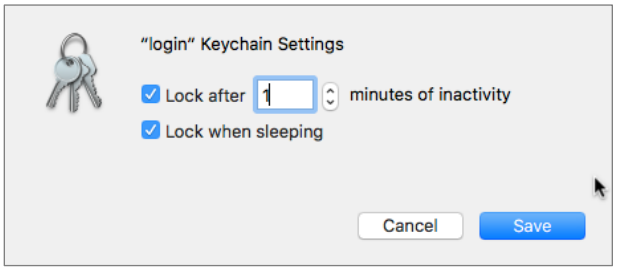
1. Open Keychain Access, located in */Applications/Utilities*. From the top of the sidebar, select the *login* keychain.



2. Select the Keychain Access *Edit* menu > *Change Settings for Keychain "login."*



3. The *Login Keychain Settings* window will open. Configure as follows:



   - Enable the *Lock after ___ minutes of inactivity* checkbox, and then set this to 1 minute.
   - Enable the *Lock when sleeping* checkbox.
4. Select the *Save* button.
5. Quit Keychain Access.
6. Sit on your thumbs for 60 seconds–time enough for the Keychain to lock.
7. Open a browser and visit a website or connect to another computer on your network that you frequent with a password that otherwise auto fills. You find you now are prompted to enter the password for the Keychain it to open.
8. If you do not need a hardened Keychain, repeat steps 1–3, and then when the *Login Keychain Settings* window appears, disable the checkboxes. Then select the *Save* button.
9. Quit Keychain Access.

Your Keychain will now automatically lock, preventing anyone from accessing all your passwords should you step away from your desk with your system awake and no screen saver in place.

4.6 Synchronize Keychain Across macOS and iOS Devices

Perhaps like me, you have a need to access most of these passwords and challenge answers anywhere, anytime. When I have my computer with me, no worries. But what if I don't? It would be a rare event indeed for me to be without my computer or my iPhone, so I keep my Keychain on my iPhone as well.

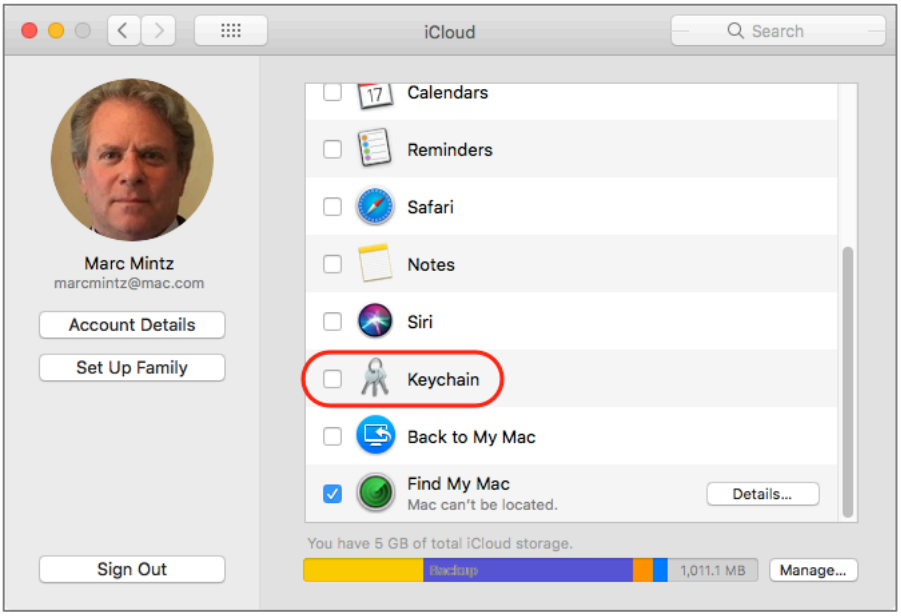
If you have upgraded to macOS 10.12 or higher, OS X 10.9 or higher, and iOS 7 or higher, Apple has you handled. With the most recent incarnations of both operating systems, Apple has added *Keychain* to the iCloud synchronization scheme. This allows your Keychain database to be synchronized between all your computers, iPhones, and iPads.

4.6.1 Assignment: Activate iCloud Keychain Synchronization

Synchronizing your Keychain with iCloud allows all your macOS 10.12 and higher, OS X 10.9 and higher, and iOS 7 and higher devices share your keychain.

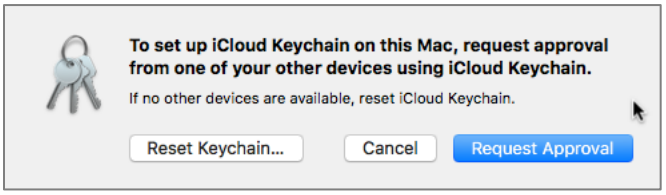
In this assignment, you enable iCloud Keychain synchronization.

1. Open the *Apple* menu > *System Preferences* > *iCloud*.

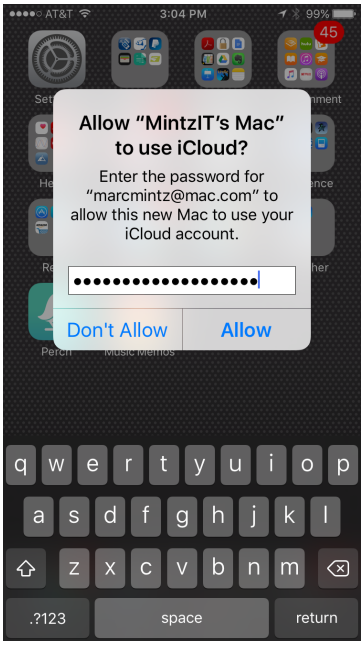


2. Select the *Keychain* checkbox. The *Enter your Apple ID password to setup iCloud Keychain* dialog box appears.
3. Enter your Apple ID password, and then select the *OK* button.

4. If you have previously created a 2-step verification for your Apple ID, the *Keychain Setup* dialog box opens. Select the *Request Approval* button.



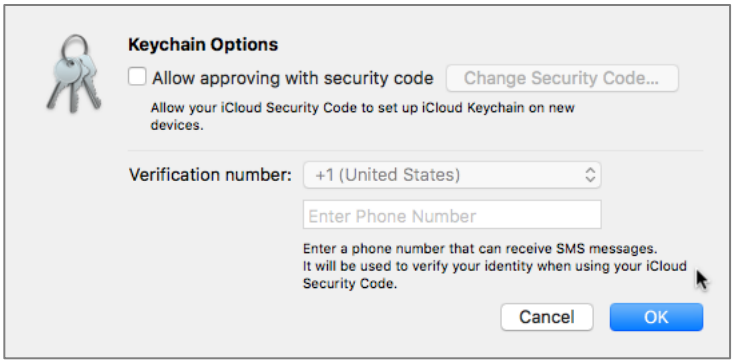
5. A request will be sent to the other devices currently approved on your account to approve this device. Enter your Apple ID password, and then click *Allow*.



6. Go back to *System Preferences,* and notice that the *Keychain* is now enabled.

Further secure your keychain:

7. In the *iCloud Preferences*, select the Keychain *Options* button.
8. The *Keychain Options* window opens:

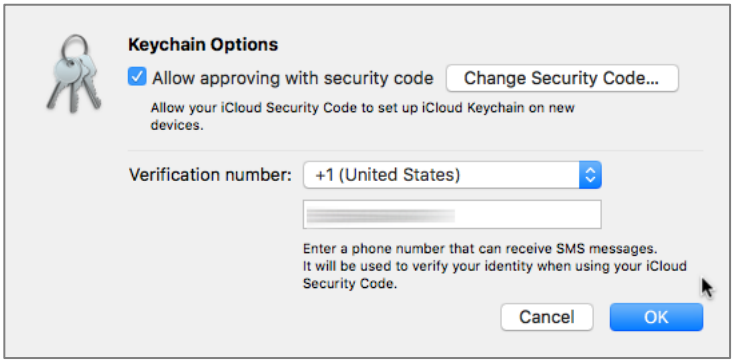


9. Enable the *Allow approving with security code* checkbox.

10. The *Create an iCloud Security Code* window opens. Enter a 6-character code that can be used to enable your other Apple devices to share and synchronize Keychains, and then select the *Next* button.
    - Notes: If you would like a more complex code, you can select the *Advanced…* button instead.

11. The same security window appears again to verify your security code. Reenter the code, and then select the *Next* button.
12. The *Enter a phone number that can receive SMS messages* window opens. This will be used by Apple to verify your identity when using the security code. Enter your phone number, and then select the *Done* button.

13. You are returned to the *Keychain Options* window. Select the *Done* button.



14. At the *Enter your Apple ID password to update your account settings* window, enter your Apple ID password, and then select the *OK* button.

15. *Quit* System Preferences.

Your Keychain on this computer will now synchronize automatically with your iCloud account, and therefore with all other OS X, macOS, and iOS devices synchronizing on the same account.

4.7 LastPass

A great solution to the problem of password management is *LastPass*[12].

There are three important advantages of LastPass:

- You no longer must concern yourself with Internet passwords–the correct response becomes automatic. LastPass will keep your Internet passwords available in each of your browsers.
- Stores and share your passwords with all your devices–even across operating systems. It also securely stores manually entered data such as challenge questions.
- The for-fee version allows sharing of selected passwords with others in the group.

LastPass provides the following solutions:

- Provides free (ad supported) and premium (no ads) options
- Automatically remembers your Internet passwords, fully encrypted
- Auto fills web-based forms and authentication fields
- Stores notes and challenge questions and answers (Q&A), fully encrypted
- Synchronizes across multiple browsers
- Synchronizes across multiple computers
- Synchronizes across Android, BlackBerry, iOS, Linux, macOS, Windows
- Automatically generates very strong passwords, which since you do not need to remember them, provide even greater online security.

4.7.1 Assignment: Install LastPass

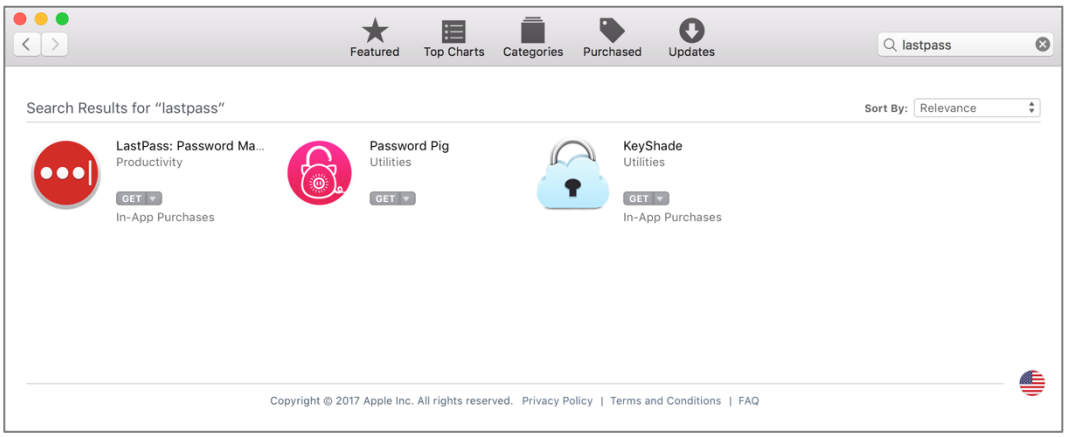
The free version of LastPass works indefinitely across devices.

[12] *http://www.LastPass.com*

In this assignment, you download and install LastPass on your macOS computer.

Download the LastPass Installer

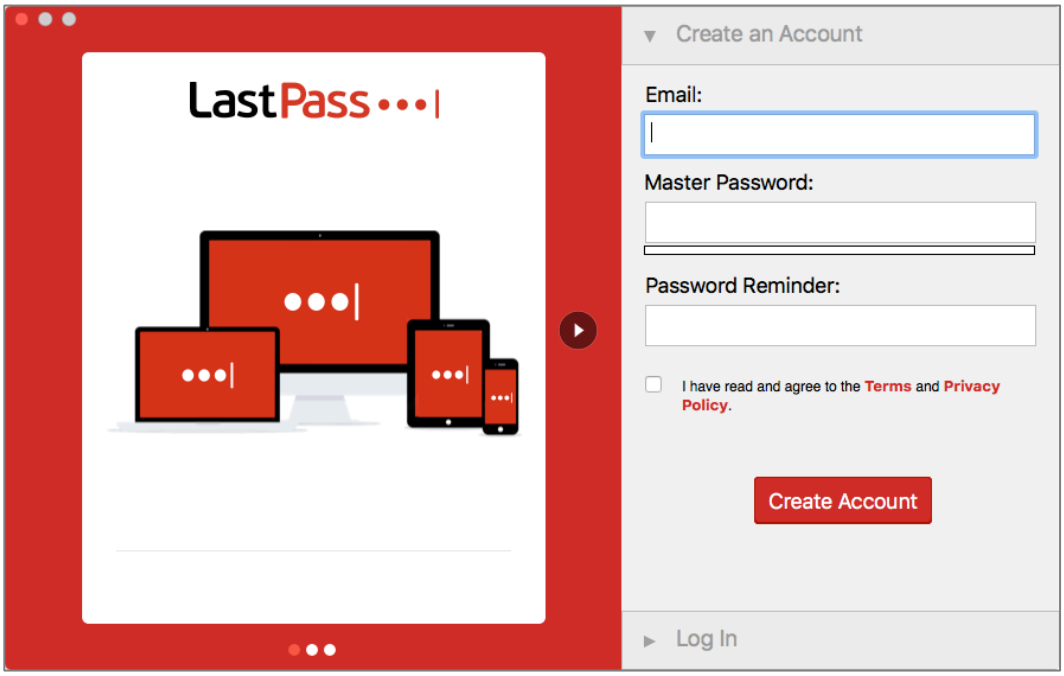
1. Open the *App Store.*
2. In the *Search Field,* enter *LastPass,* and then tap the *Return/Enter* key.
3. In the *LastPass* area, select *Get.* LastPass will download.



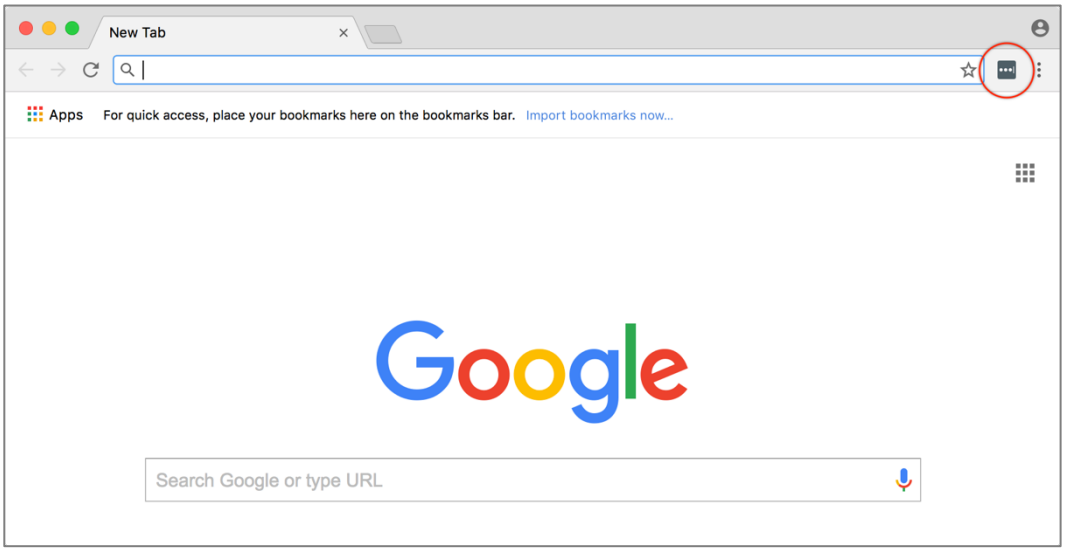
Install LastPass

4. Once LastPass has downloaded, double-click to launch it.

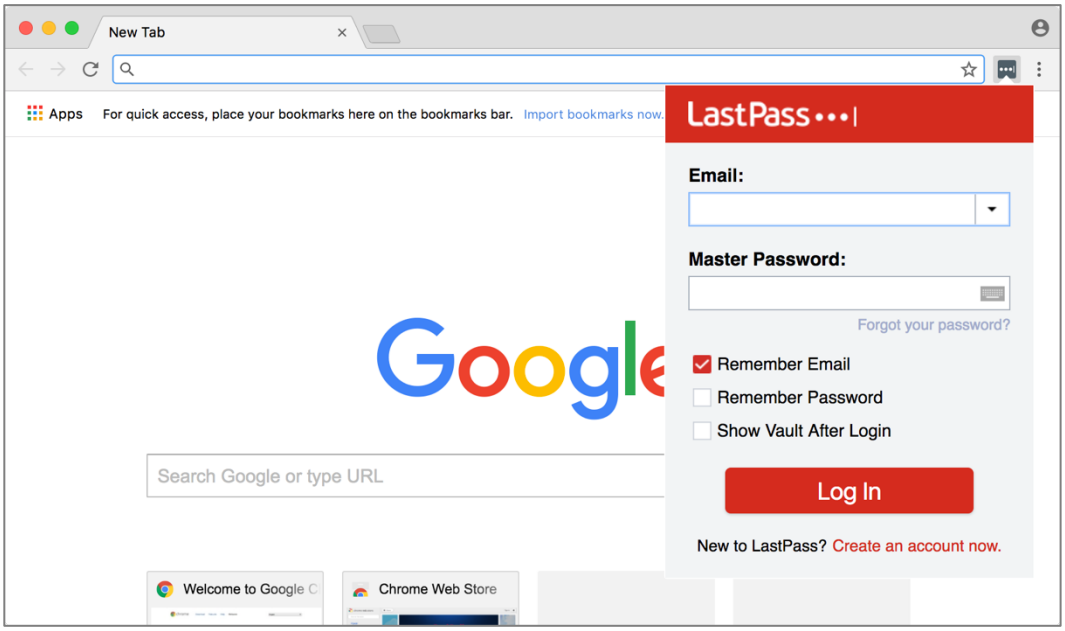
5. Select *Create an Account,* and then enter your *Email* address, a password in the *Master Password* field, a *Password Reminder*, and then click *Create Account*.



6. LastPass automatically installs its extension into Chrome, Edge, Internet Explorer, Firefox, Opera, and Safari. Open a browser. In this example, it is Chrome. The LastPass extension displays as three dots …



7. In your browser, click the LastPass extension icon. The LastPass window opens.



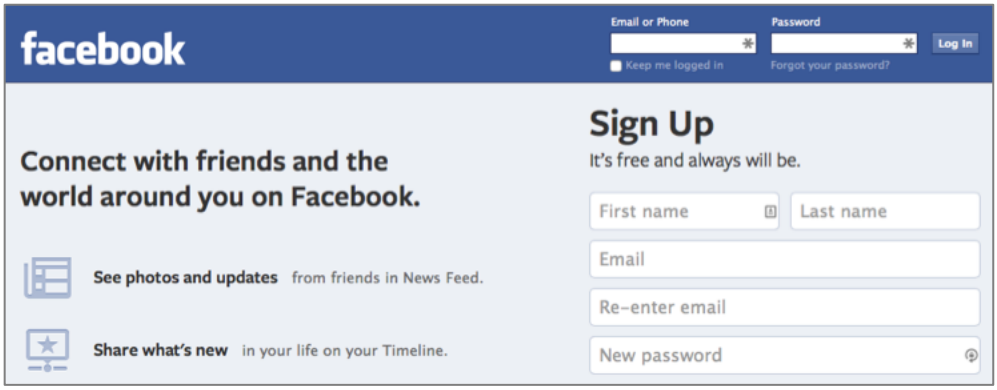
8. Enter the *email* address to be linked to LastPass, and then the *Master Password* you created in an earlier step, and then click *Log In*.
9. The LastPass window goes away, and LastPass is now active within your browser.
10. If you use multiple browsers, repeat steps 6-9 with each.

4.7.2 Assignment: Use LastPass to Save Website Authentication Credentials

Once you have LastPass installed, it's time to put it to use.

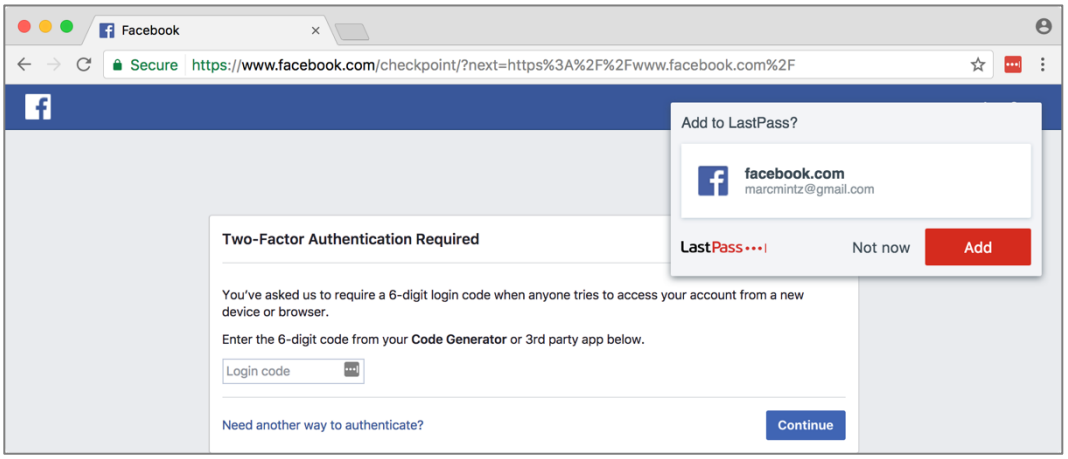
In this assignment, you use LastPass to store the user name and password for Facebook.

1. Use your browser to visit Facebook *https://facebook.com*.



2. As this is the first time you have visited Facebook since installing LastPass, your log in credentials have not yet been stored in LastPass. Enter your Email or Phone and Password information, and then select the *Log in* button.

3. LastPass will detect that there is a form on this page, and present an option to remember your credentials. This will appear just under the navigation bar. Select the *Add* button.



4. Quit your web browser.

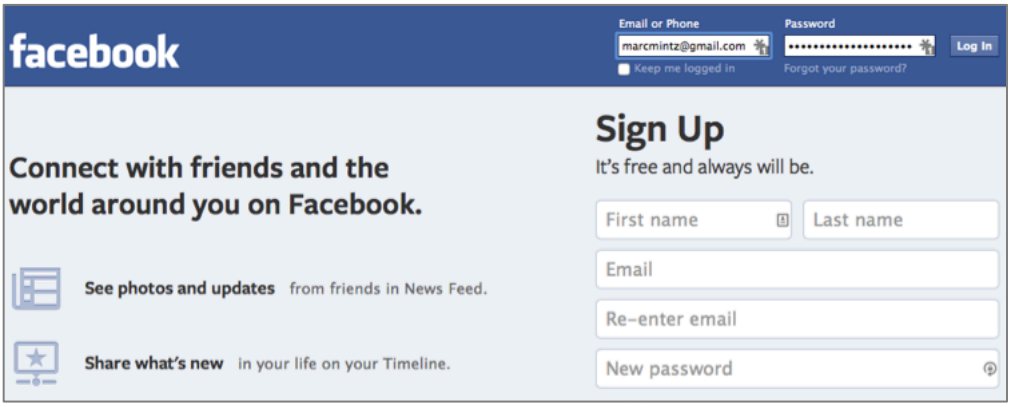
Your Facebook account credentials are now stored in LastPass, so you do not need to remember them.

4.7.3 Assignment: Use LastPass to Auto Fill Website Authentication

When LastPass has saved user name and password information for a site, you will never need to manually enter that information again.

In this assignment, you revisit Facebook and allow LastPass to enter your credentials.

1. Launch your browser and then go to *Facebook* at *https://facebook.com*. Take note that your authentication credentials have been automatically entered for you by LastPass.



2. Quit your browser.

You have just successfully proved that LastPass is saving your credentials.

4.8 Password Policies

Within the government, military, financial, and healthcare environments, setting *password policies* is often a regulatory mandate. Although not a mandate for the home and general business computer, doing so makes a lot of sense.

A password policy is a set of rules to help users create and use passwords. You have likely seen password policies in use when creating a password for your online banking or shopping, and were alerted that your password needed to be longer, or have a special character.

In an IT environment which is controlled by either a Microsoft Active Directory or macOS Server, password policies can be enforced from the server. Within environments without a server, you can enforce password policies using either the Terminal for command line control, or the macOS Server app for graphical control. In the following exercise, even though your computer is not in an environment controlled by a server, you will install and configure macOS server to manage password policies on a computer.

4.8.1 Assignment: Password Policies with macOS Server

The primary difference between the Mac computer you are using, and a Mac server is the installation of the Apple Server app. The server app is available from the App Store for $19.95. Compared to the time and energy required to properly configure password policies through the command line, this is a bargain.

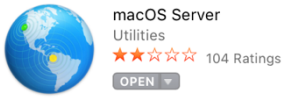
Should you be feeling particularly nerdy, open the *Terminal.app*, enter *man pwpolicy,* and then tap the *Return/Enter* key. *pwpolicy* is the command-line method of setting password policy in macOS. Although pwpolicy still works under macOS 10.13, it is mostly depreciated. The modern method of controlling password policies is with *profile keys*[13].

[13] *https://developer.apple.com/library/content/featuredarticles/iPhoneConfigurationProfileRef/Introduction/Introduction.html*

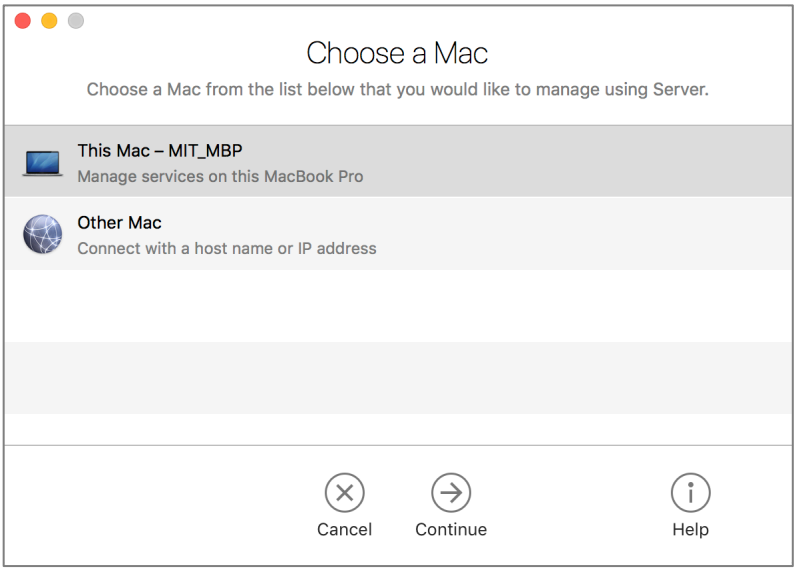
In this assignment, you will install and configure macOS Server app to manage password policies on your computer, for all users of your computer.

Install macOS Server app

1. Open *Apple* menu > *App Store.*
2. In the *search* field, enter *Server,* and then tap the *Enter* or *Return* key.
3. Click the *macOS Server* icon.

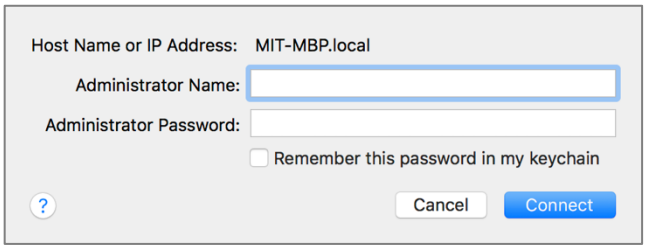


4. Purchase macOS Server app.
5. Once macOS Server app has downloaded to your computer, double-click to open it (located in the */Applications* folder).
6. At the *Choose a Mac* window, select *This Mac,* and then click the *Continue* button at the bottom center of the window.

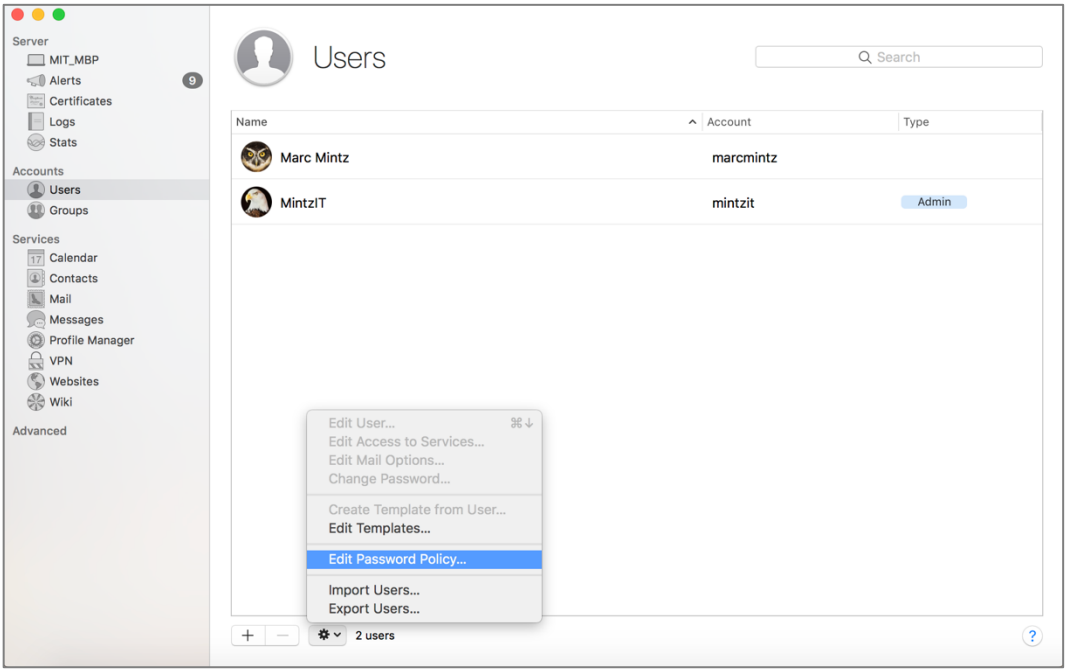


7. At the authentication window, enter and administrator's name and password, and then click the *Connect* button.

    - NOTE: *Do not* enable *Remember this password in my keychain.* This will help prevent unauthorized users from accessing the server app.

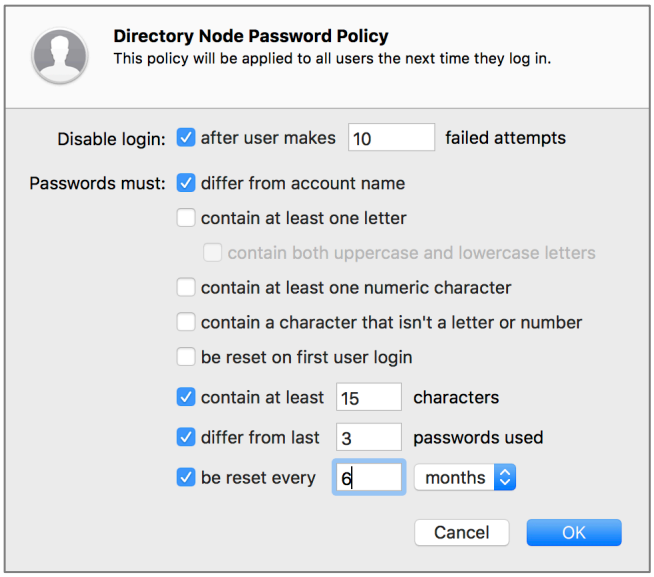


8. Select *Users* in the sidebar, click the *gear* icon > *Edit Password Policy…*



9. In the *Directory Node Password Policy* window, configure to your taste.

    - NOTE: Based on the definition of *Strong Password* used in this book, and the loosening requirement for frequent password changes, you may want to configure your policies as below:



10. Click *Ok* button, and then *Quit* Server app.
11. Restart the computer to implement the change.

Test the Password Policy

12. Once logged back into your computer, open *Apple* menu > *System Preferences* > *Users & Groups*.
13. Try to change your own password, using one that does not you're your new password policy. Notice how you are alerted and that you must follow policy.
14. Cancel the password change.
15. Authenticate as an administrator.
16. Create a new user account, and attempt to assign it a password that does not meet your new password policy. Notice how you are alerted and that you must follow policy.
17. Cancel creating a new user.
18. Exit System Preferences.

Revision Log

20180420, v2.0

- The majority of chapters have been edited for updated information.
- *Chapter 2.6 renumbered for readability.*
- *Chapter 4.5.1 Assignment: Harden the Keychain with a Different Password* removed. As of macOS 10.13.4 the login keychain password cannot be changed from the user account login password.
- Chapter *19.3 NordVPN* revised to create a free trial account.
- *Chapter 20.3 Facebook* heavily edited to reflect the revised privacy and timeline settings.
- *Chapter 20.4 LinkedIn* heavily edited to reflect the revised privacy settings.
- *Chapter 20.5 Google* heavily edited to reflect the revised privacy and Takeout options.

20180325, v 1.3

- *Chapter 4.8 Password Policies* added.
- *Chapter 12.1 Find My Mac* has been slightly edited.
- *Chapter 14.8 Do Not Track* has been edited to reflect changes in Ghostery, and the Chrome extension installation process.
- *Chapter 15.7 End-To-End Secure Email With GNU Privacy Guard* rewritten to reflect the major update of GPGTools.
- *Chapter 19.3 NordVPN* is rewritten from scratch from our previous recommended VPN host.

20171022, v1.2

- *Chapter 14 Web Browsing* is rewritten.

- *Chapter 15 Email,* added *hacked-emails.com* for checking if your email account was included in site breaches.
- *Chapter 16 Apple ID and iCloud*, added that Two-Factor Authentication can use either text messaging or voice call.
- *Chapter 19 Internet Activity,* changed the recommended VPN provider to *Perfect-Privacy.com.*

20171001, v1.1

- Updated chapter *Documents > Encrypt A Folder for Cross Platform Use With Zip* to use Keka, instead of the depreciated macOS built-in tools.

20170923, v1.01

- Updated chapter *When It Is Time To Say Goodbye*

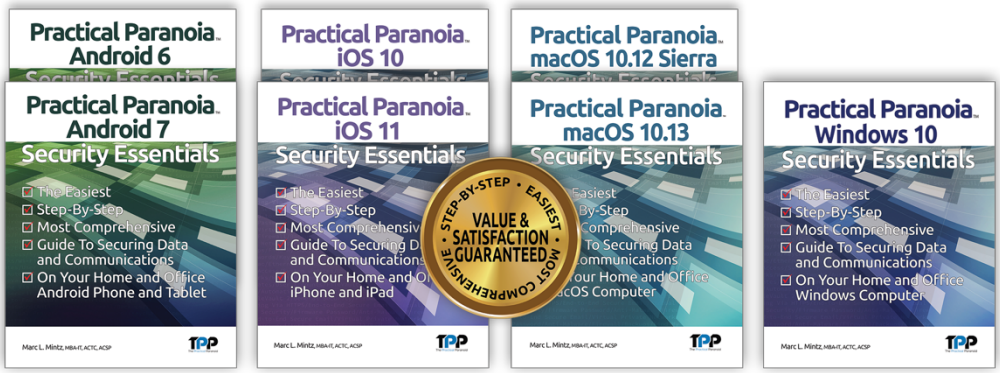
20170918, v1.0

Initial release

Index

Practical Paranoia Workshops & Books

4 Years Undisputed #1 Best, Easiest, & Most Comprehensive Cybersecurity Series



This is an age of government intrusion into every aspect of our digital lives, criminals using your own data against you, and teenagers competing to see who can crack your password the fastest. Every organization, every computer user, everyone should be taking steps to protect and secure their digital lives.

The *Practical Paranoia: Security Essentials Workshop* is the perfect environment in which to learn not only *how*, but to actually *do* the work to harden the security of your macOS and Windows computers, and iPhone, iPad, and Android devices.

Workshops are available online and instructor-led at your venue, as well as tailored for on-site company events.

Each Book is designed for classroom, workshop, and self-study. Includes all instructor presentations, hands-on assignments, software links, and security checklist. Available from Amazon (both print and Kindle format), and all fine booksellers, with inscribed copies available from the author.

Call for more information, to schedule your workshop, or order your books!

The Practical Paranoid, LLC
+1 888.504.5591 • info@thepracticalparanoid.com
https://thepracticalparanoid.com