

---

# Harden the World

*Release 0.1*

Jul 01, 2018



---

## Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Application Hardening . . . . .	3
1.2	Operating System Hardening . . . . .	66
<b>2</b>	<b>Contributing</b>	<b>123</b>
2.1	Feedback . . . . .	123
<b>3</b>	<b>License</b>	<b>125</b>





Every day many people, including me, waste time googling for procedures, configurations and a way to harden their services. So I created Harden the World in the hope to start a community project focused on developing common guidelines and best practices to deploy secure configurations. This repository contains hardening guidelines for devices, applications and OSs.

**Project home:** <http://hardentheworld.org>

**Project repository:** <https://github.com/jekil/hardentheworld>



## 1.1 Application Hardening

This chapter describe how to harden standalone applications. It is divided in two sections: client side applications (i.e. browser, email client) and server side applications (i.e. web server, file server).

### 1.1.1 Apple Mail 8

Apple Mail is a mail client application shipped by default with Max OS X.

This chapter is dedicated to configuring Apple Mail version 8.x. It comes by default with Mac OS X 10.10 (Yosemite).

- *Disable automatic account settings*
- *Disable automatic attachment download*
- *Disable automatic loading of remote content*
- *Disable MailDrop*
- *Never add invitations to calendar automatically*
- *Never add invitations to calendar automatically*
- *Use only SSL/TLS protocols*
- *Using GPG*

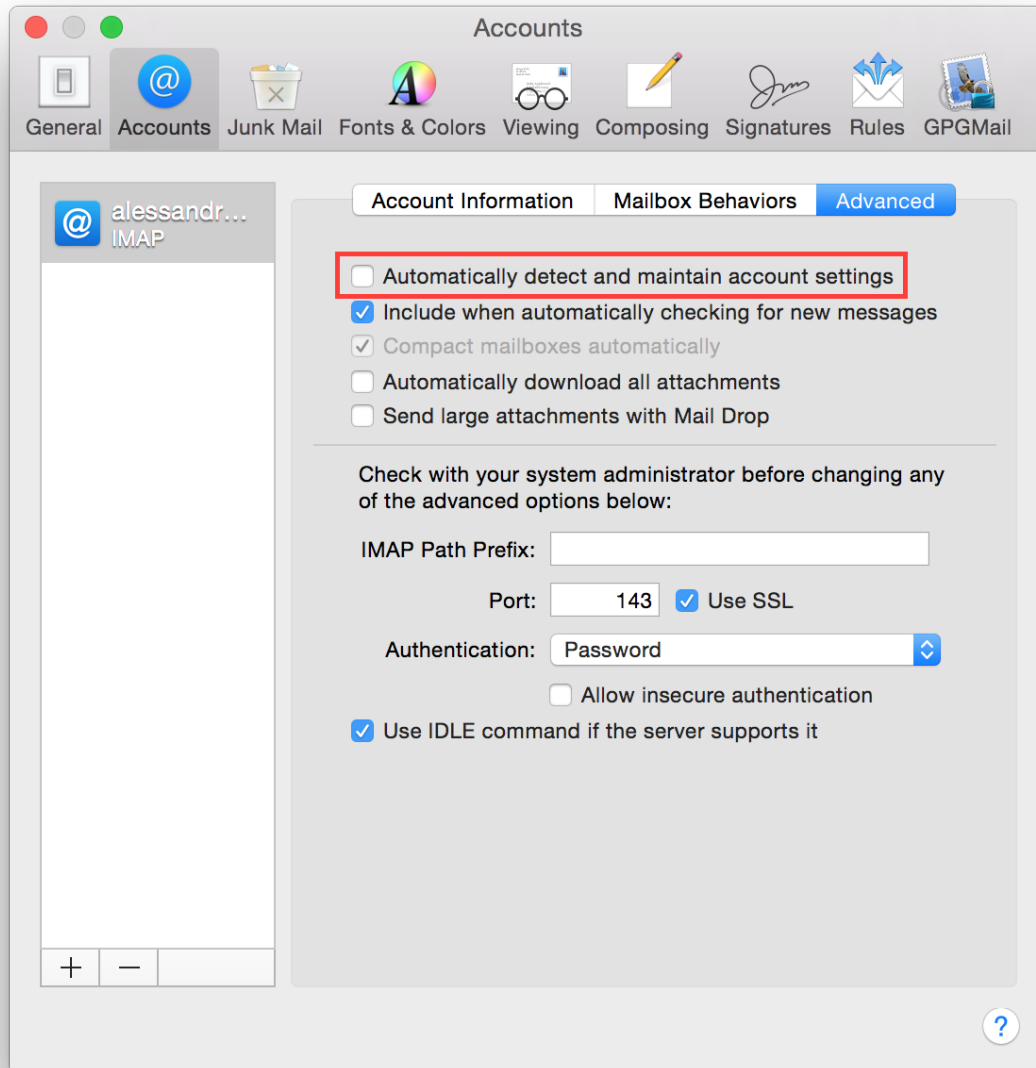
#### Disable automatic account settings

If this options is enabled Mail automatically manage settings for your email account in Mail, such as port numbers and authentication methods. It is not suggested to not leave Mail the control over so critical settings and disable this option.

To disable automatic account settings, go to:

Open Apple Mail Accounts Select your mail account Advanced

Uncheck “Automatically detect and maintain account settings”.

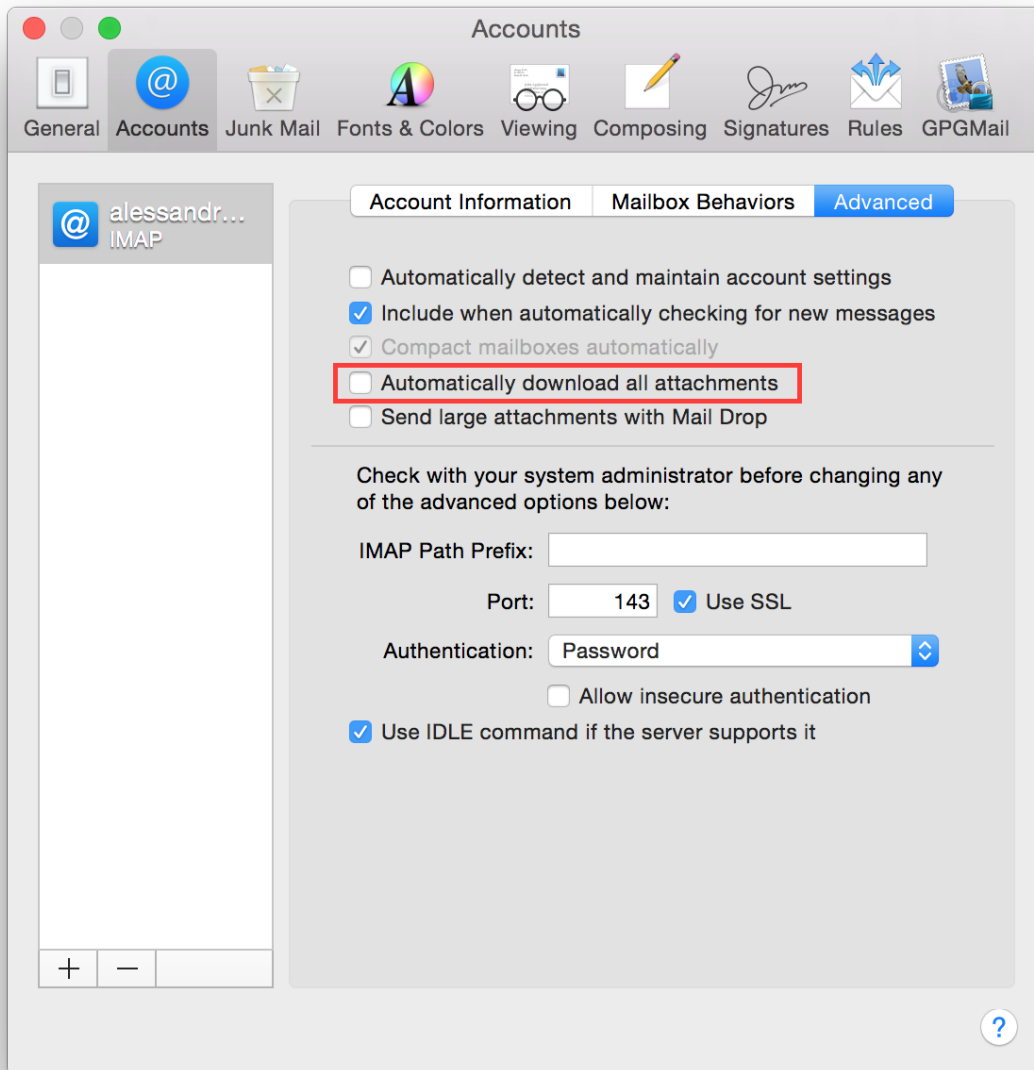


### Disable automatic attachment download

If this options is enabled Mail automatically downloads all attachments for your email account in Mail. It is suggested to keep the control over what is downloaded so disable this option, automatically download attacchments is pretty dangerous, just think to someone sending you an email with an image on a controlled server, he could be able to track your IP address.

It is suggested to disable automatic attachments download, go to:

Open Apple Mail Accounts Select your mail account Advanced  
Uncheck “Automatically download all attachments”.



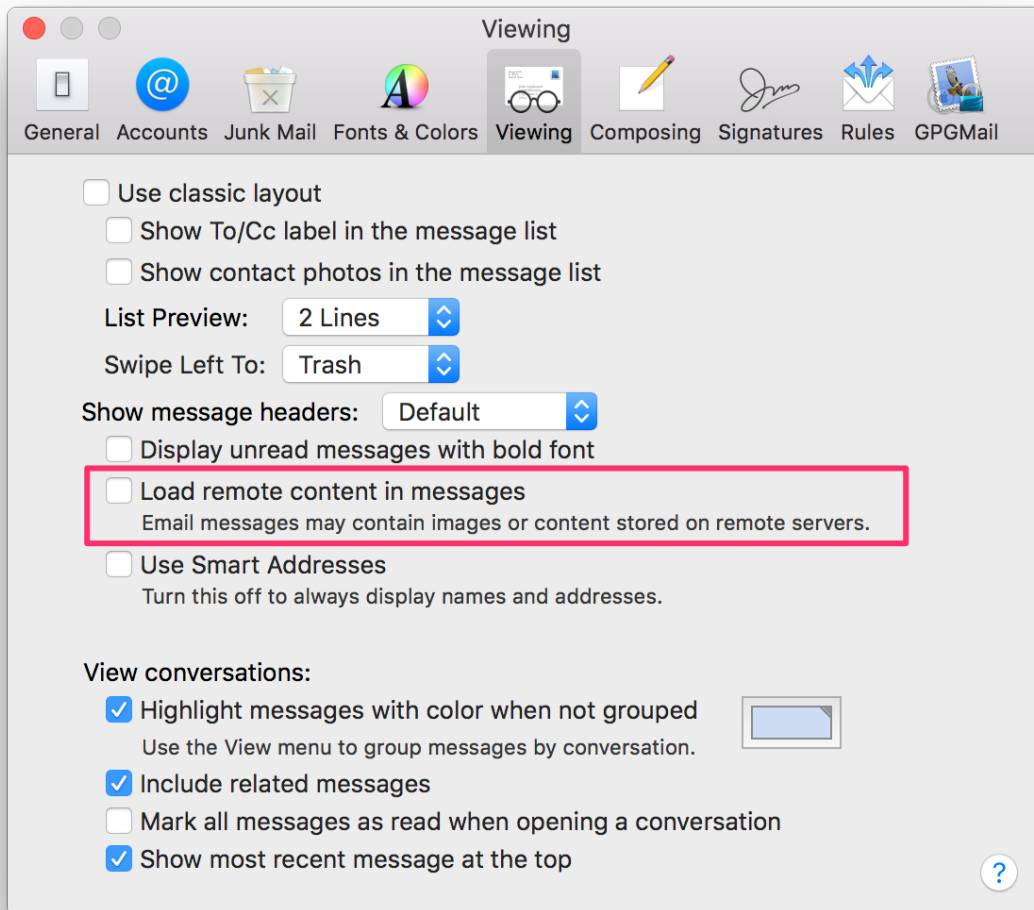
### Disable automatic loading of remote content

Mail defaults to automatically load any images, styles etc, that are included in any email, regardless of sender. Not only can this be an attack-vector, but it's also commonly used for tracking, leading to loss of privacy.

Don't worry about disabling the automatic loading though, you'll still be able to load remote images and stylesheets for any mail with a single click.

To disable automatic loading of remote content, go to:

Open Apple Mail Preferences Viewing  
Uncheck “Load remote content in messages”.



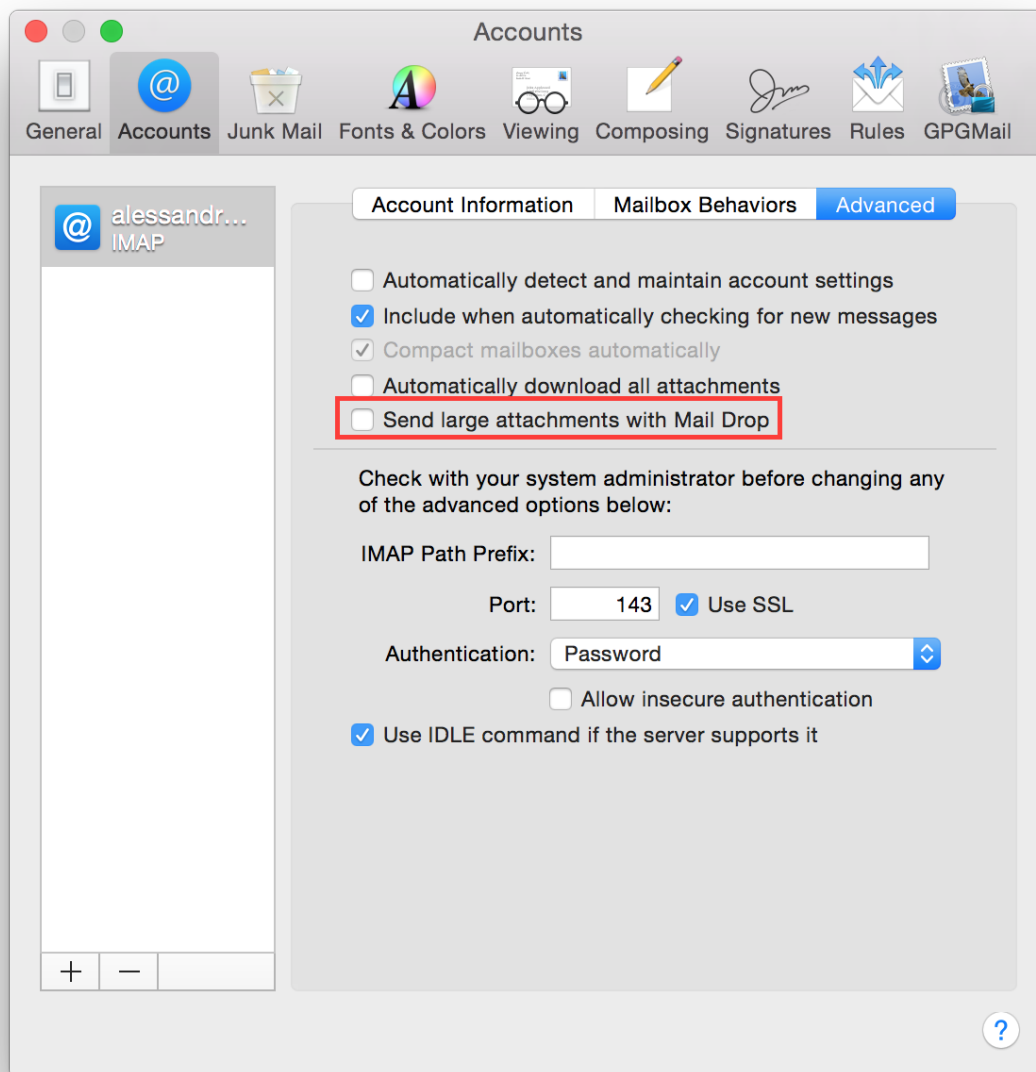
If you want to definitely block any connections it is suggested to configure a firewall, i.e. Little Snitch, and permit connections starting from Mail.app only to your mail server.

### Disable MailDrop

MailDrop is a new feature in Yosemite which allows you to deliver large size attachments, they are uploaded to Apple Cloud and then fetched by your recipients. This is a great feature but it needs to disclose your file to Apple Cloud. It is suggested to disable this feature and use other technology under your full control to transfer big files.

To disable invitation import, go to:

Open Apple Mail Accounts Select your mail account Advanced  
Uncheck “Send large attachments with Mail Drop”.



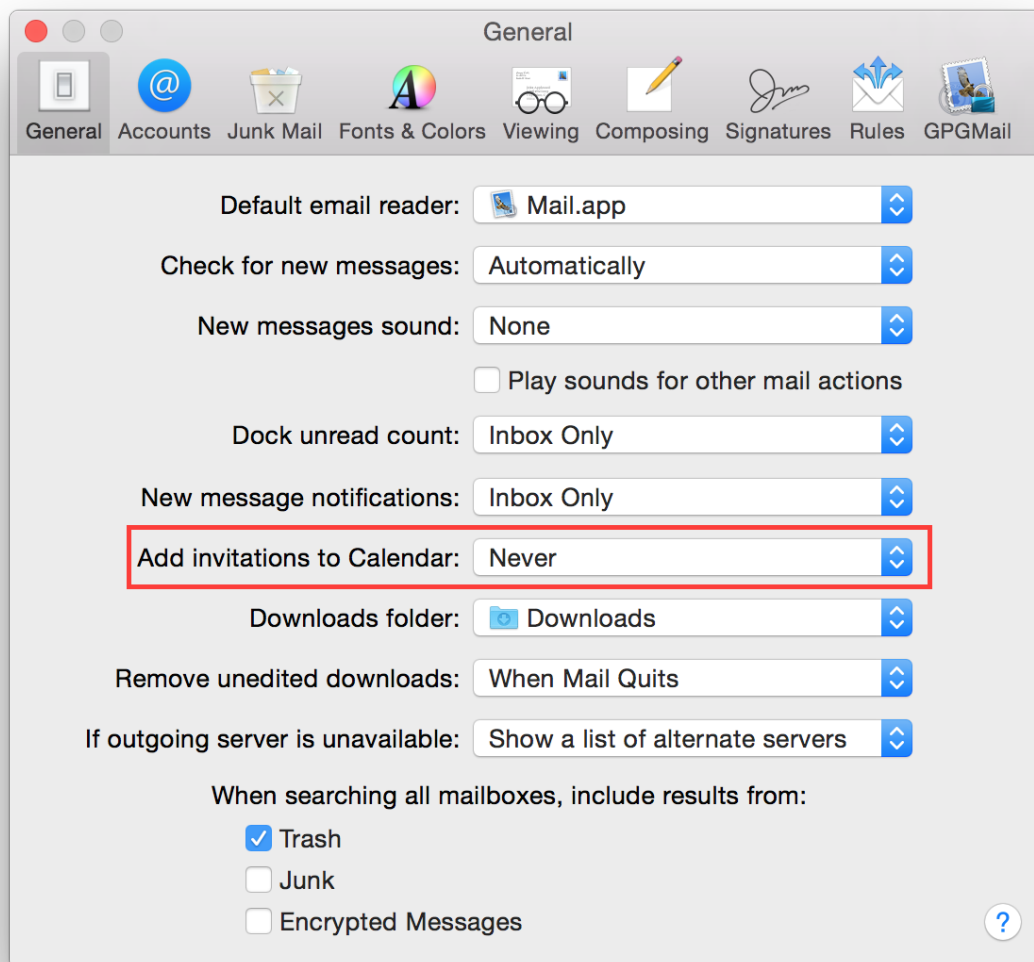
### Never add invitations to calendar automatically

Apple Mail has the feature to automatically add invitations to your calendar. It is suggested to not allow Apple Mail to automatically parse invitations and launch an external application to avoid possible future exploitation with a new vulnerability.

To disable invitation import, go to:

Open Apple Mail General

Set “Add invitations to Calendar” to “Never”.



### Never add invitations to calendar automatically

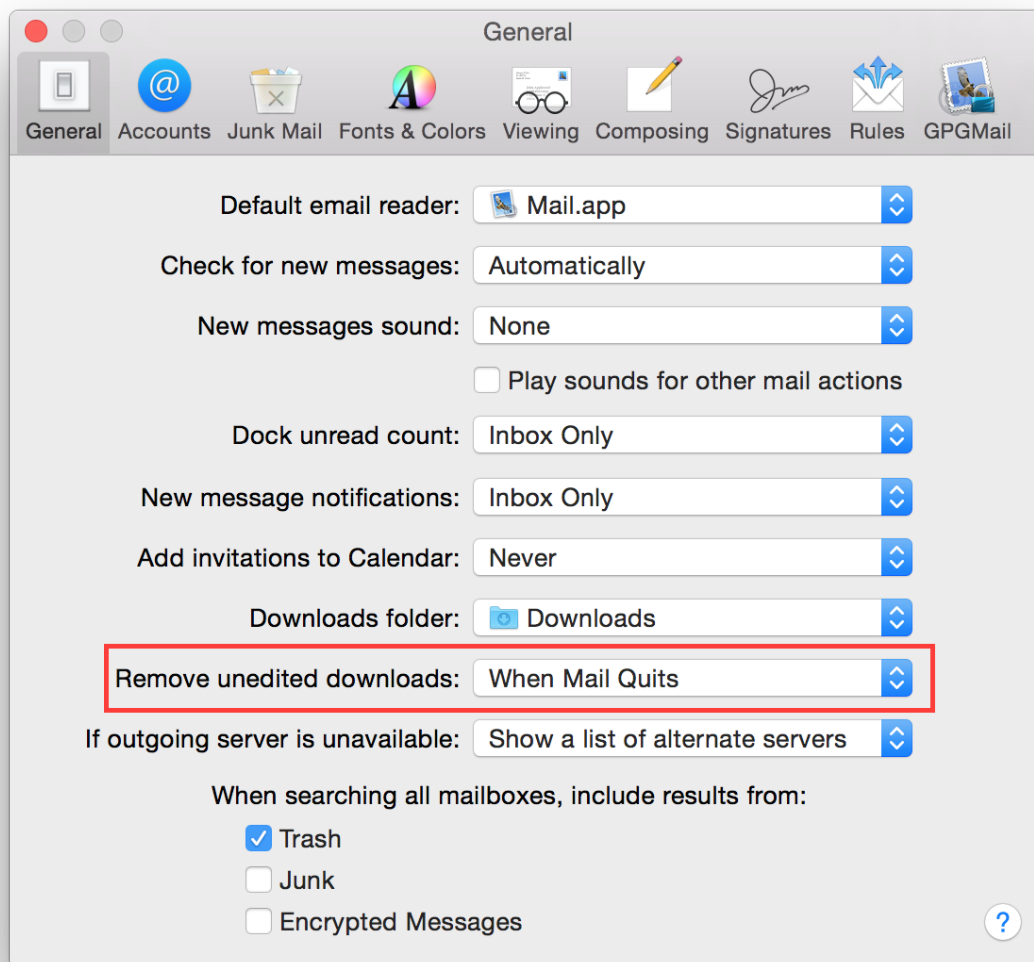
If you open an attachment in Apple Mail, it stores the file in your disk, inside your attachment download folder, and by default leaves it forever. It is not suggested to leave mail attachments on disk, because they can be some kind of untrusted files.

To remove downloaded files, go to:

Open Apple Mail General

Set “Remove unedited downloads” to “When Mail Quits”.





### Use only SSL/TLS protocols

Classic mail protocols like SMTP, POP and IMAPS are plain text protocol without any encryption, it means your data and credentials are send in plain text. It is suggested to use only encrypted protocols. Ask your email provider for encrypted email protocols support and configure your mail account properly.

To configure your email account, go to:

Open Apple Mail Accounts

### Using GPG

GPG is a software to encrypt, decrypt, sign and verify files or messages. It is widely used and its adoption is suggested to protect your privacy.

[GPGTools](#) is a suite designed to bring GPG on Mac OS X and add encryption to Apple Mail.

It is suggested to download and install [GPGTools](#).

### 1.1.2 Apple Mail 9

Apple Mail is a mail client application shipped by default with Mac OS X.

This chapter is dedicated to configuring Apple Mail version 9.x. It comes by default with Mac OS X 10.11 (El Capitan).

- *Disable automatic account settings*
- *Disable automatic attachment download*
- *Disable automatic loading of remote content*
- *Disable MailDrop*
- *Never add invitations to calendar automatically*
- *Never add invitations to calendar automatically*
- *Use only SSL/TLS protocols*
- *Using GPG*

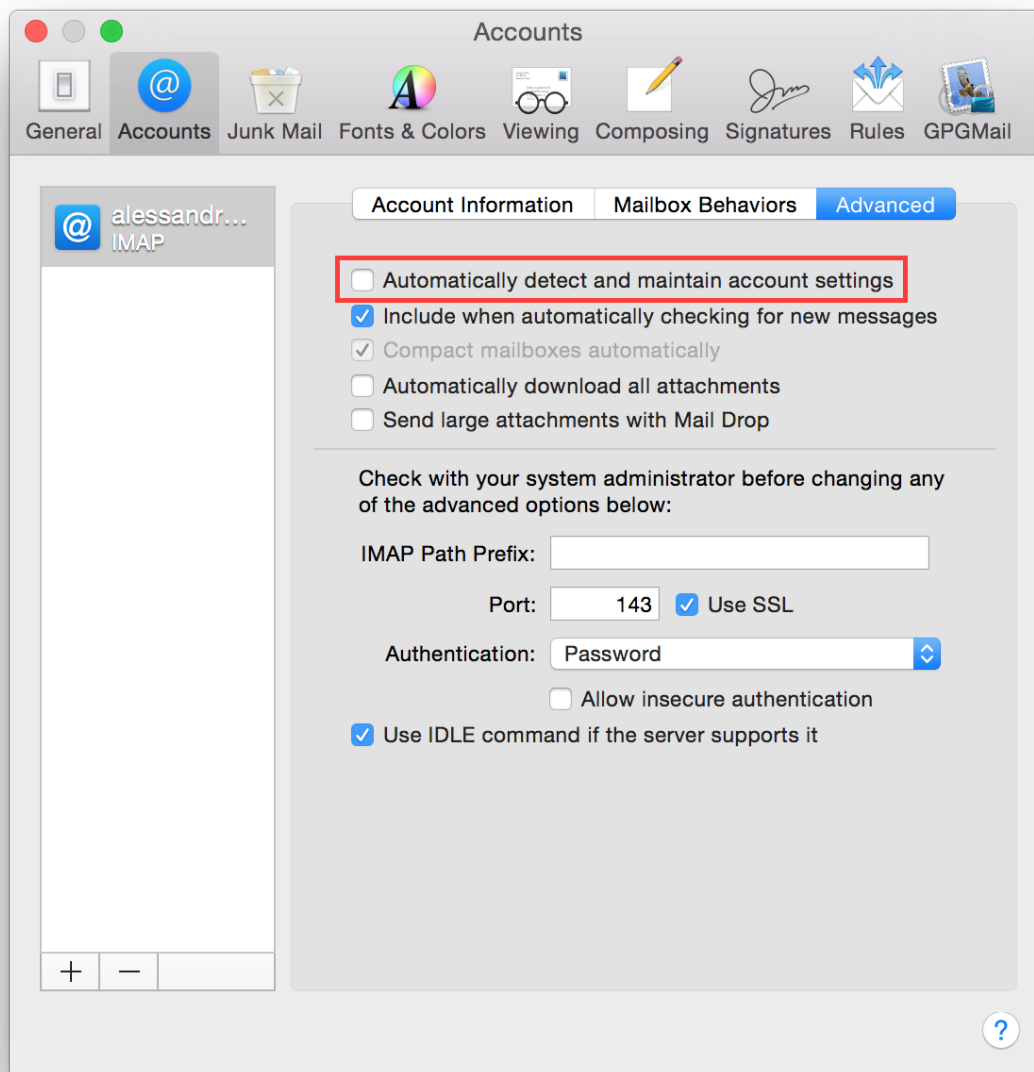
#### Disable automatic account settings

If this options is enabled Mail automatically manage settings for your email account in Mail, such as port numbers and authentication methods. It is not suggested to not leave Mail the control over so critical settings and disable this option.

To disable automatic account settings, go to:

Open Apple Mail Accounts Select your mail account Advanced

Uncheck “Automatically detect and maintain account settings”.



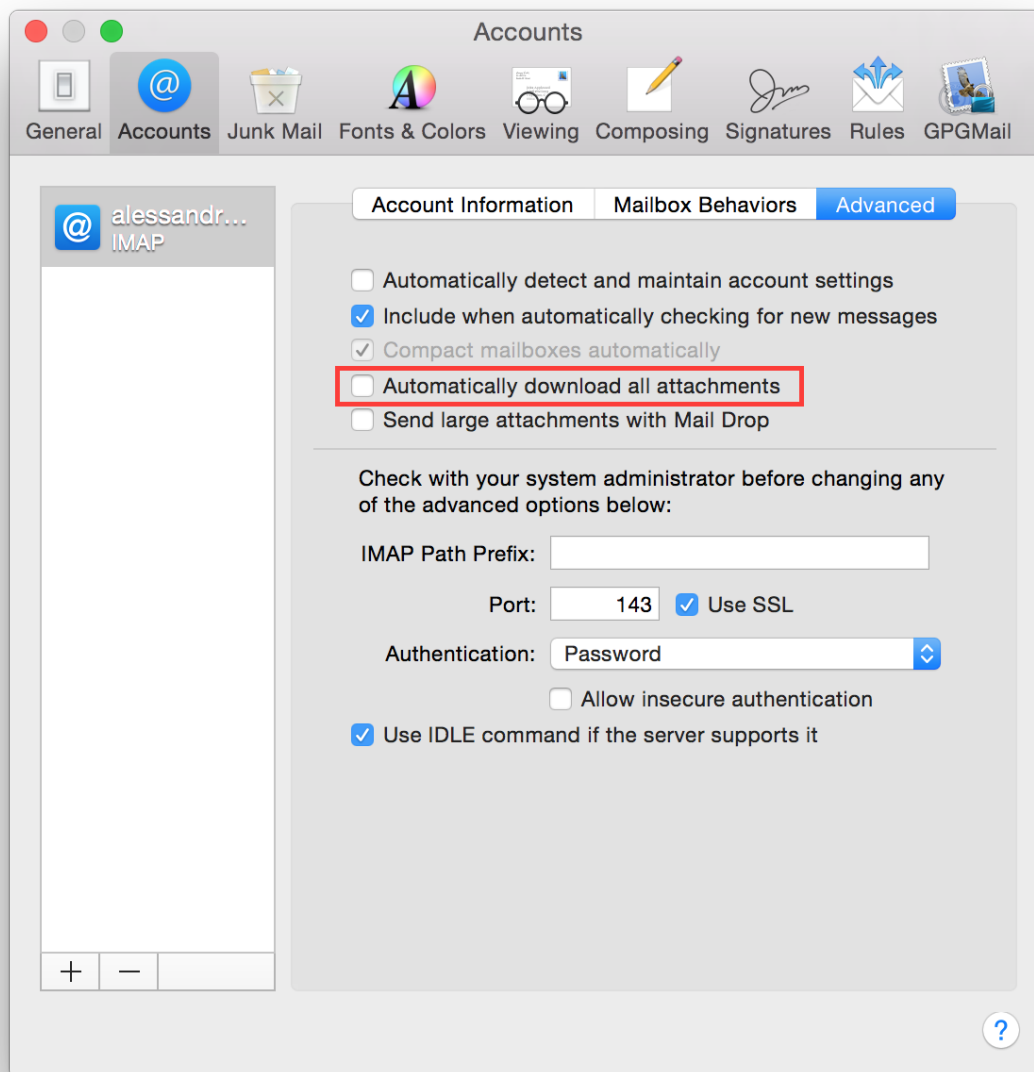
### Disable automatic attachment download

If this options is enabled Mail automatically downloads all attachments for your email account in Mail. It is suggested to keep the control over what is downloaded so disable this option, automatically download attachments is pretty dangerous, just think to someone sending you an email with an image on a controlled server, he could be able to track your IP address.

It is suggested to disable automatic attachments download, go to:

Open Apple Mail Accounts Select your mail account Advanced

Uncheck “Automatically download all attachments”.



## Disable automatic loading of remote content

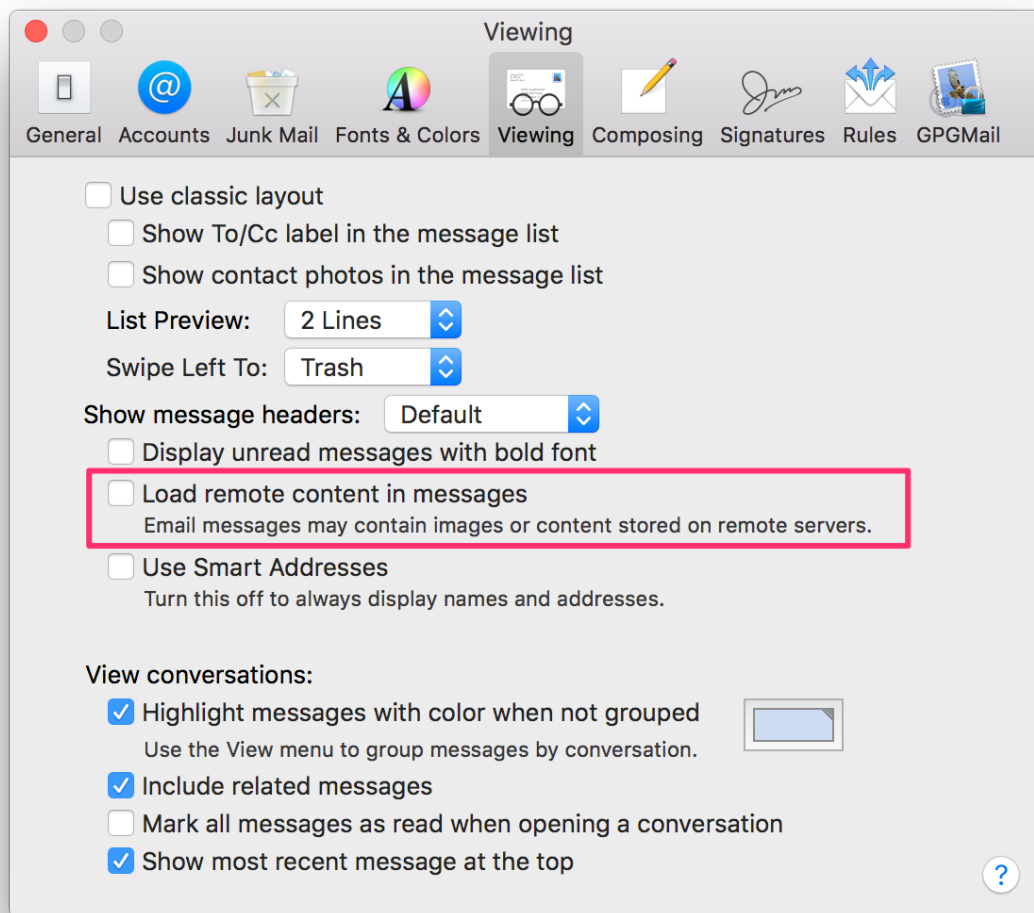
Mail defaults to automatically load any images, styles etc, that are included in any email, regardless of sender. Not only can this be an attack-vector, but it's also commonly used for tracking, leading to loss of privacy.

Don't worry about disabling the automatic loading though, you'll still be able to load remote images and stylesheets for any mail with a single click.

To disable automatic loading of remote content, go to:

Open Apple Mail Preferences Viewing

Uncheck "Load remote content in messages".



If you want to definitely block any connections it is suggested to configure a firewall, i.e. Little Snitch, and permit connections starting from Mail.app only to your mail server.

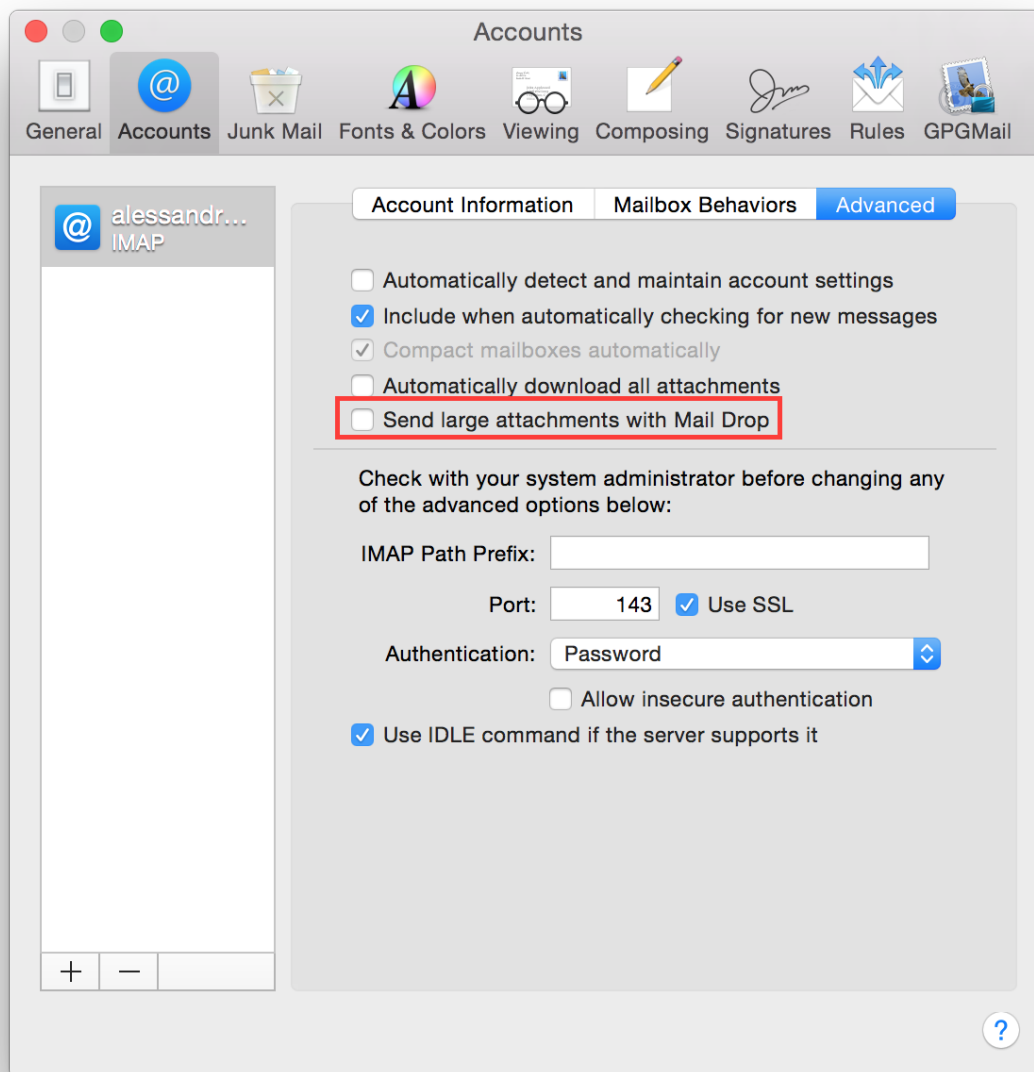
### Disable MailDrop

MailDrop is a new feature in Yosemite which allows you to deliver large size attachments, they are uploaded to Apple Cloud and then fetched by your recipients. This is a great feature but it needs to disclose your file to Apple Cloud. It is suggested to disable this feature and use other technology under your full control to transfer big files.

To disable invitation import, go to:

Open Apple Mail Accounts Select your mail account Advanced

Uncheck "Send large attachments with Mail Drop".



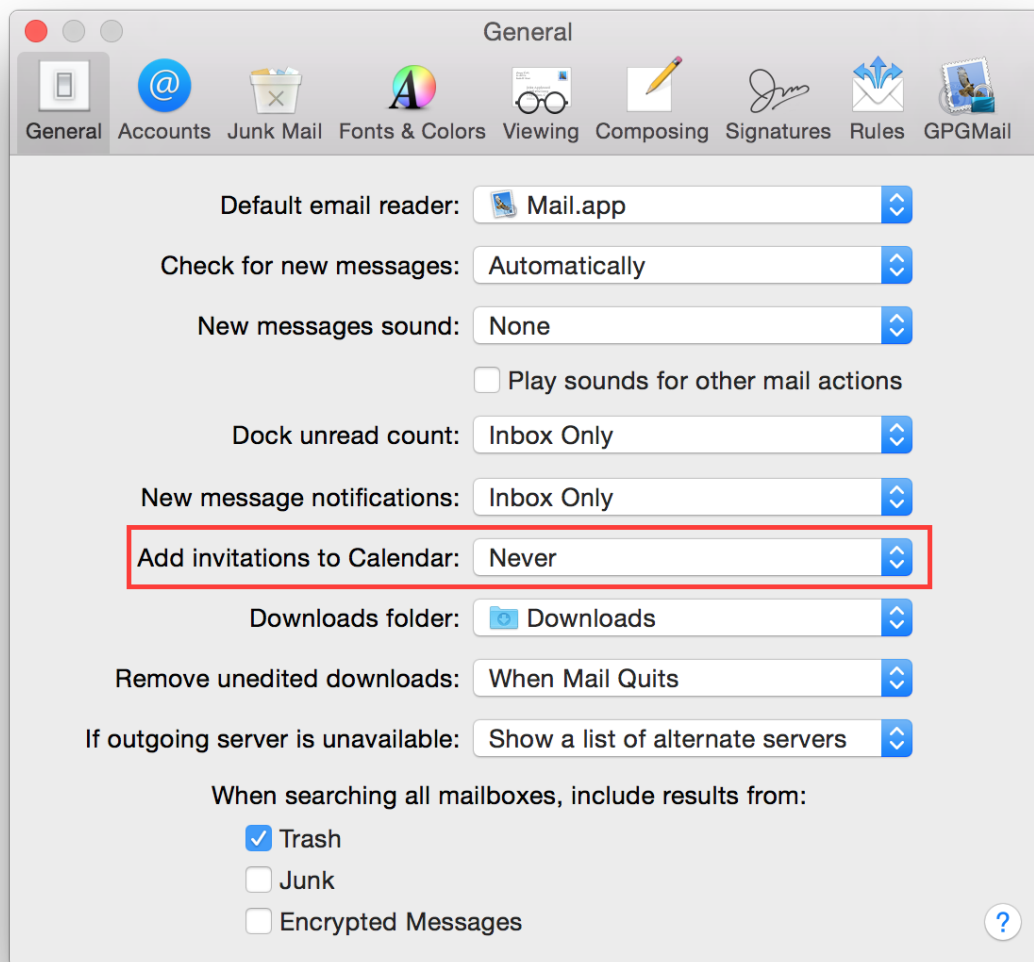
### Never add invitations to calendar automatically

Apple Mail has the feature to automatically add invitations to your calendar. It is suggested to not allow Apple Mail to automatically parse invitations and launch an external application to avoid possible future exploitation with a new vulnerability.

To disable invitation import, go to:

Open Apple Mail General

Set “Add invitations to Calendar” to “Never”.



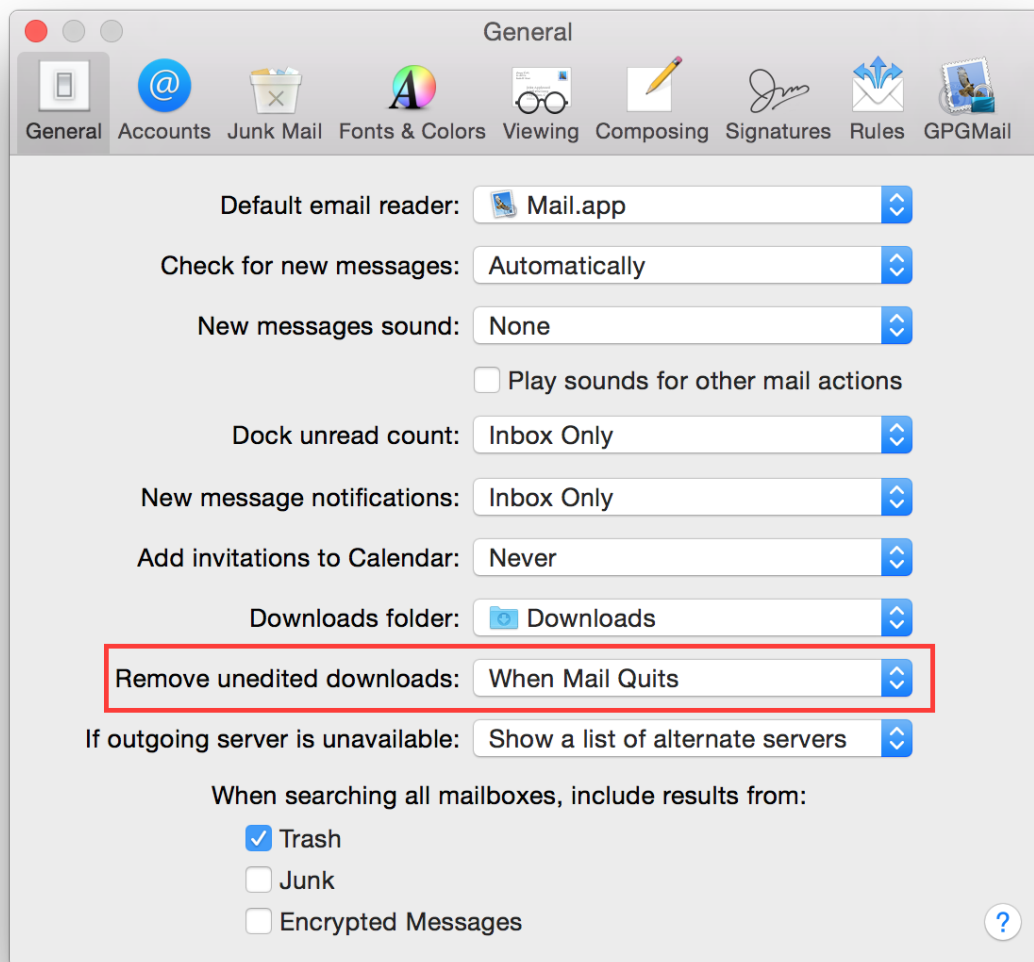
### Never add invitations to calendar automatically

If you open an attachment in Apple Mail, it stores the file in your disk, inside your attachment download folder, and by default leaves it forever. It is not suggested to leave mail attachments on disk, because they can be some kind of untrusted files.

To remove downloaded files, go to:

Open Apple Mail General

Set “Remove unedited downloads” to “When Mail Quits”.



## Use only SSL/TLS protocols

Classic mail protocols like SMTP, POP and IMAPS are plain text protocol without any encryption, it means your data and credentials are send in plain text. It is suggested to use only encrypted protocols. Ask your email provider for encrypted email protocols support and configure your mail account properly.

To configure your email account, go to:

Open Apple Mail Accounts

## Using GPG

GPG is a software to encrypt, decrypt, sign and verify files or messages. It is widely used and its adoption is suggested to protect your privacy.

[GPGTools](#) is a suite designed to bring GPG on Mac OS X and add encryption to Apple Mail.



It is suggested to download and install [GPGTools](#).

### 1.1.3 Apple Safari 8

According to [Wikipedia](#) Safari is “a web browser developed by Apple Inc. included with the OS X and iOS operating systems. First released as a public beta on January 7, 2003, on the company’s OS X operating system, it became Apple’s default browser beginning with Mac OS X v10.3 “Panther”. The native browser of iOS is also called Safari, but has a different UI and uses a different WebKit version and API”.

This chapter is dedicated to configuring Apple Safari version 8.x. It comes by default with Mac OS X 10.10 (Yosemite).

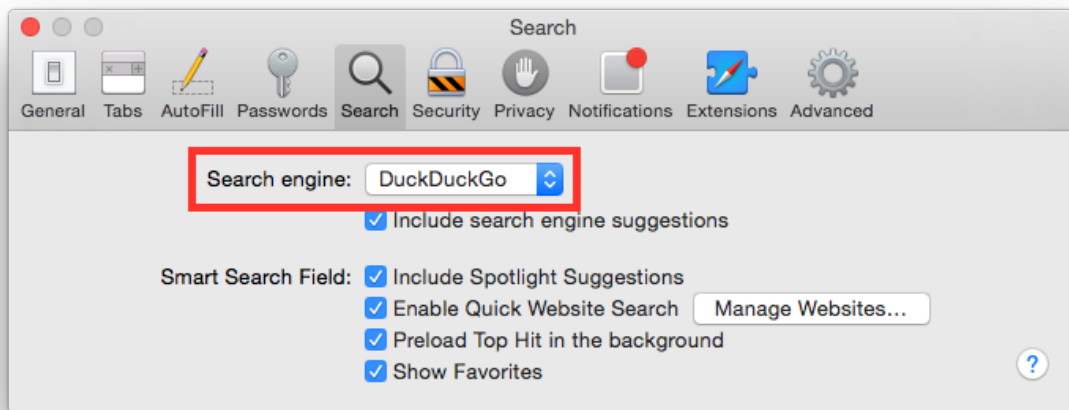
- *Change default search engine*
- *Clear history*
- *Control third party plugins*
- *Disable AutoFill*
- *Disable AutoFill username and passwords*
- *Disable open files after download*
- *Disable search suggestions*
- *Disable website tracking*
- *Open with a private window*
- *Open with an empty page*
- *Show website address*
- *Warn when visiting a fraudulent website*

#### Change default search engine

Change default search engine to [DuckDuckGo](#), it is a search engine who takes care of your privacy. Go to:

Open Safari Preferences Search

Set “Search engine” to “DuckDuckGo”.

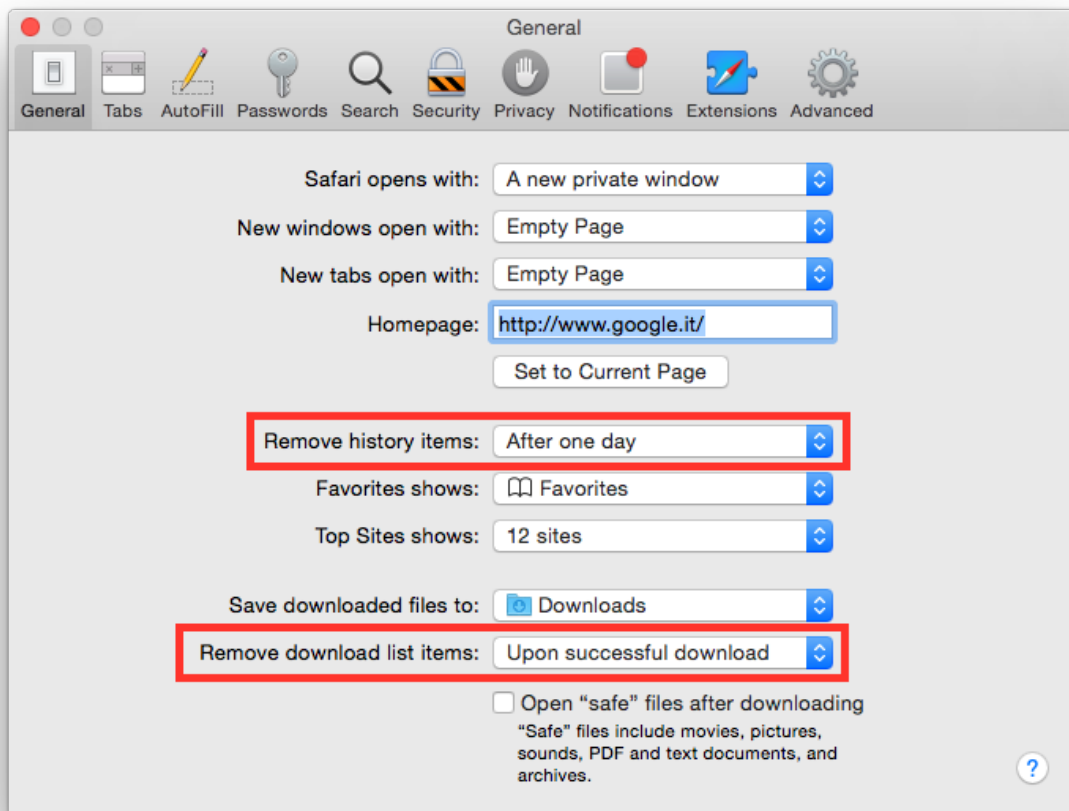


### Clear history

Used to never leave navigation traces in browser history. Go to:

Open Safari Preferences General

Set "Remove history items" to "After one day". Set "Remove download list items" to "Upon successful download".

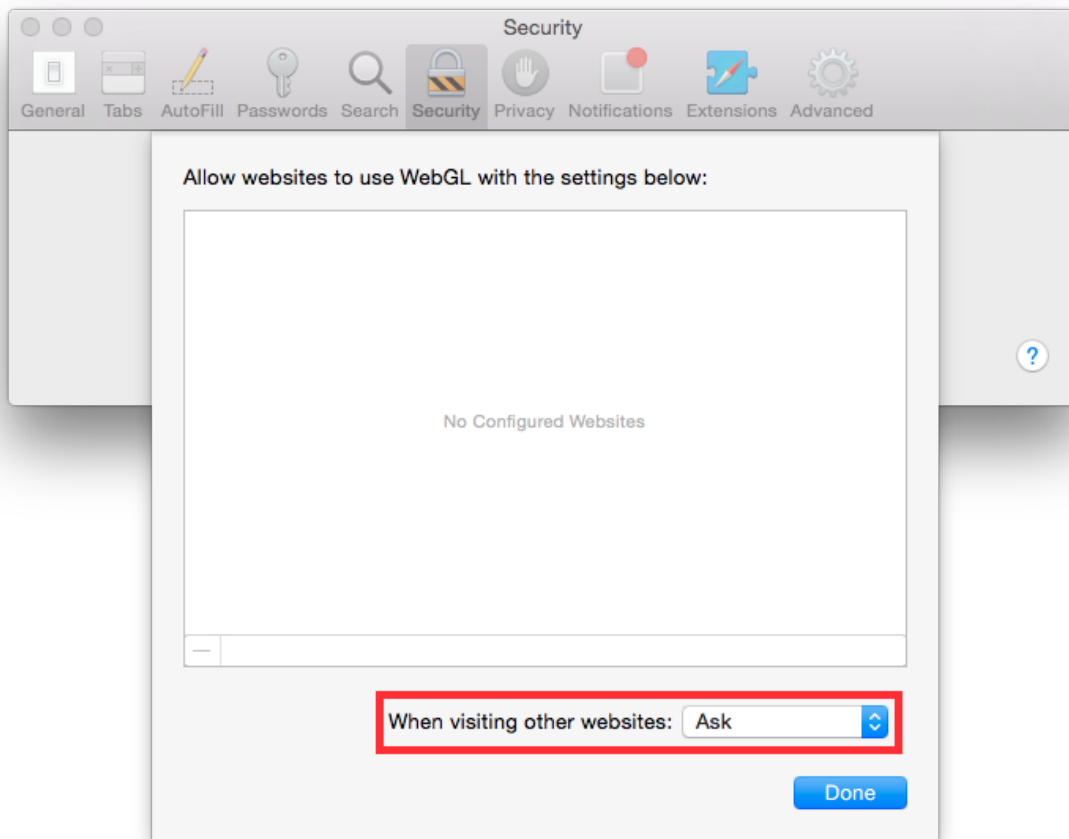


## Control third party plugins

Most browsers allow the continuous running of all third party scripts, giving malware an huge surface area of attack to get into your machine. Safari can ask for user permission each time a plugin is run, this is a good practice to control which website is asking your browser to run a plugin. It is suggested to control the run of WebGL plugins, go to:

Open Safari Preferences Security

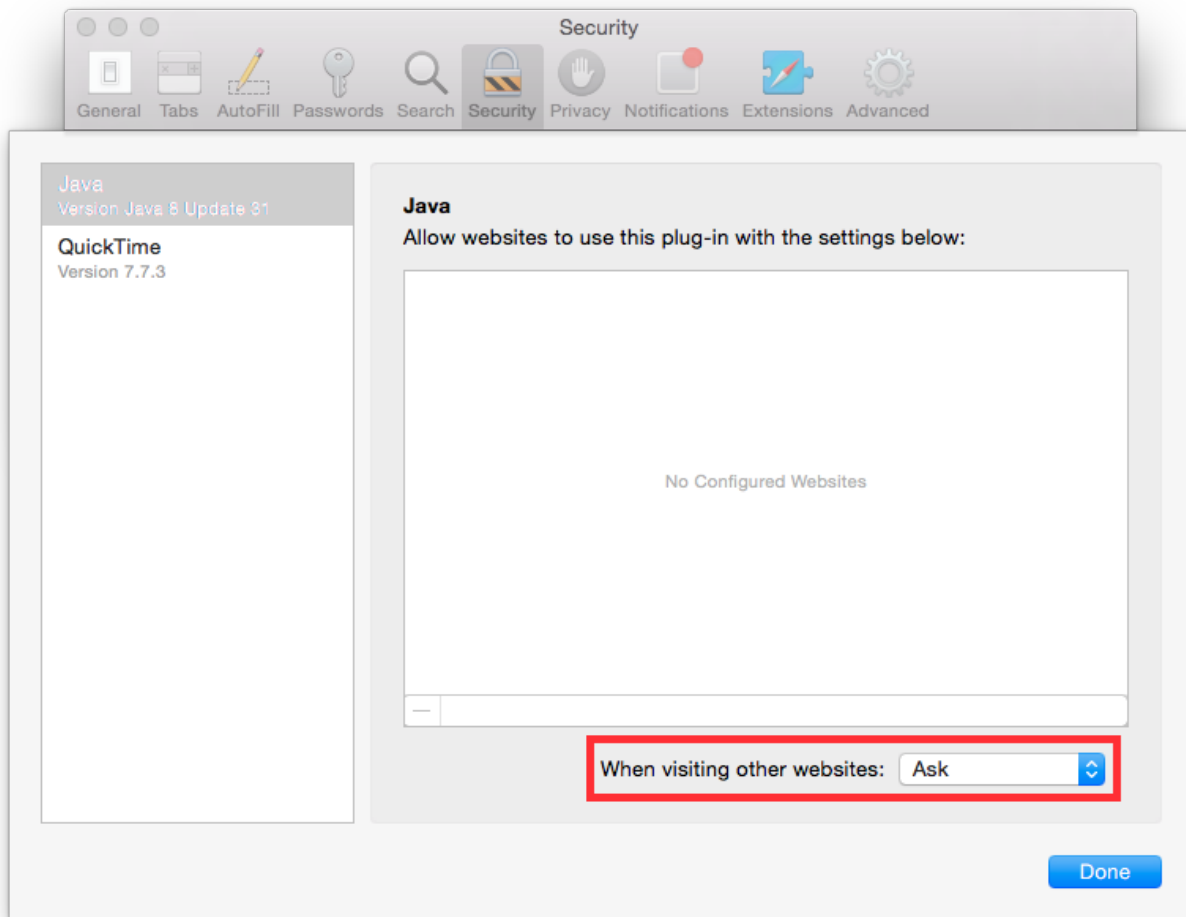
Check "Allow WebGL" and click on "Website Settings...", set "When visiting other websites" to "Ask".



It is suggested to control the run of WebGL plugins, go to:

Open Safari Preferences Security

Check “Allow Plug-ins” and click on “Website Settings. . .”, set “When visiting other websites” to “Ask”.

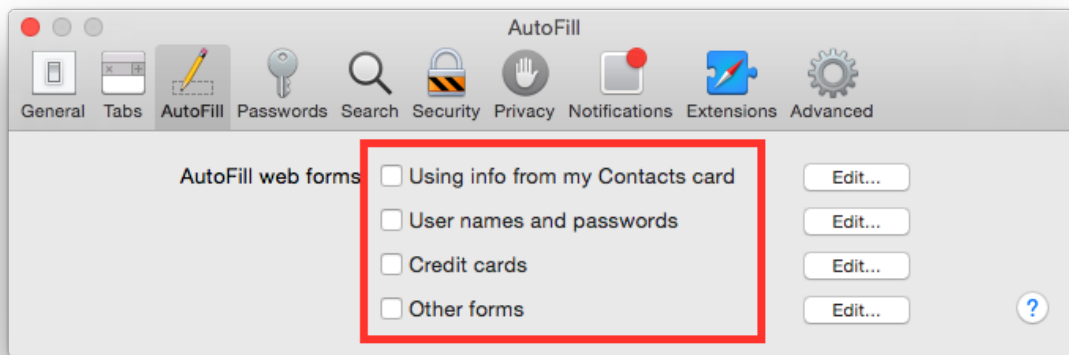


### Disable AutoFill

Disables automatic fill of forms. Go to:

Open Safari Preferences AutoFill

Uncheck all boxes.

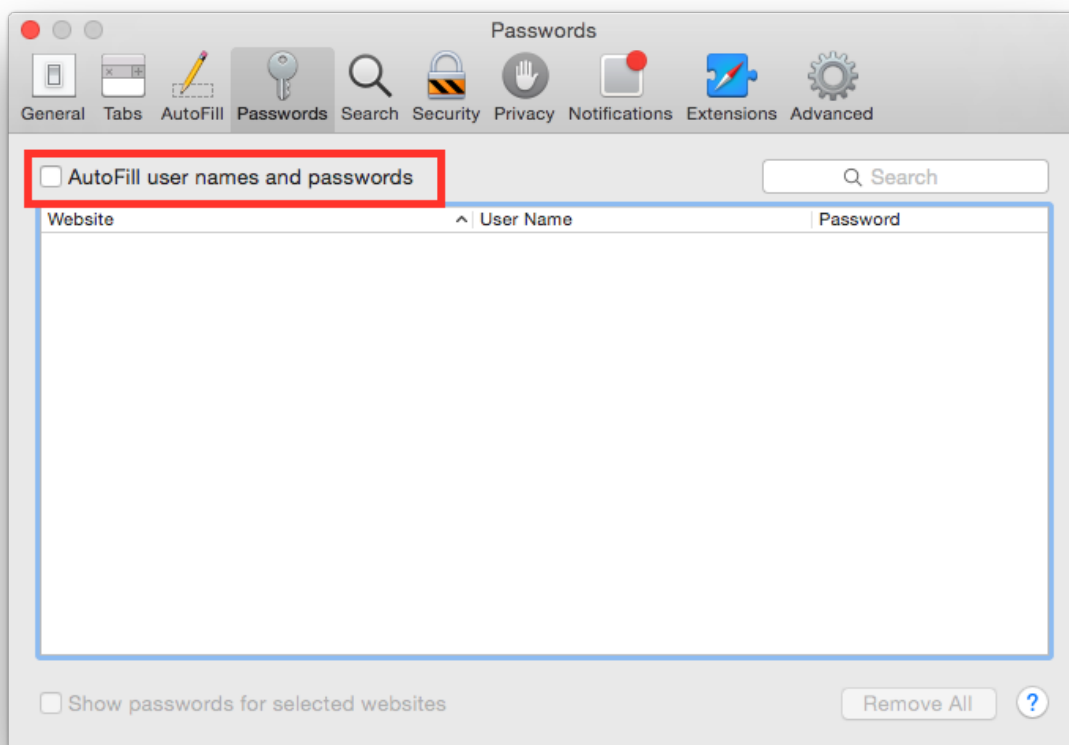


### Disable AutoFill username and passwords

Disables automatic fill of forms with usernames and passwords. Go to:

Open Safari Preferences Passwords

Uncheck "AutoFill user names and passwords".

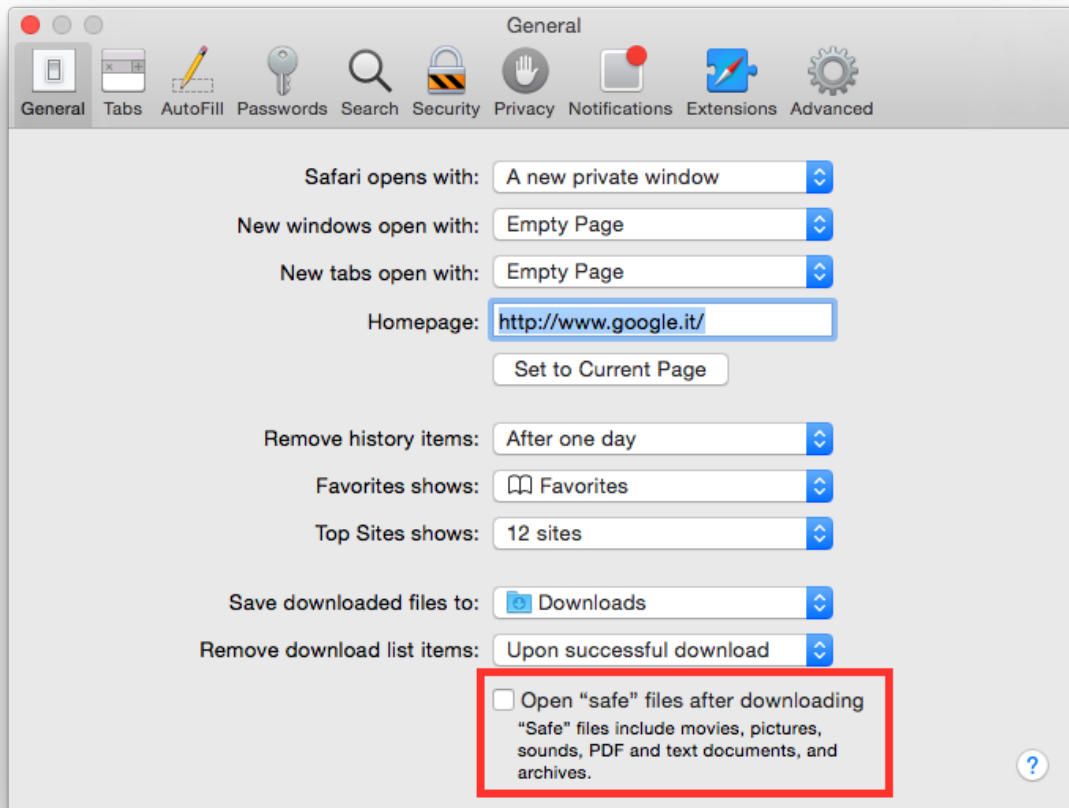


## Disable open files after download

Disables automatic opening of downloaded file, even if they are safe. It is suggested to never run arbitrary files downloaded. Go to:

Open Safari Preferences General

Uncheck “Open ‘safe’ files after downloading”.

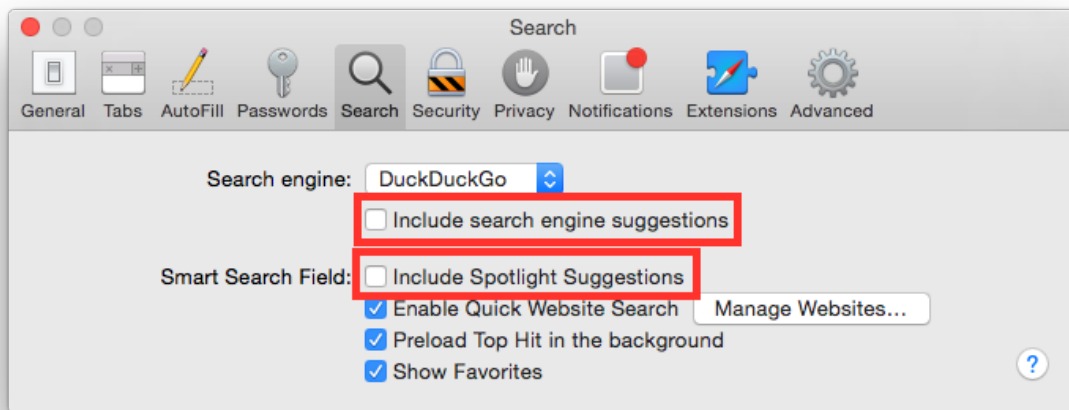


## Disable search suggestions

Disables suggestions to avoid leaking potential data when typing in the search box. Go to:

Open Safari Preferences Search

Uncheck “Include search engine suggestions”. Uncheck “Include Spotlight suggestions”.

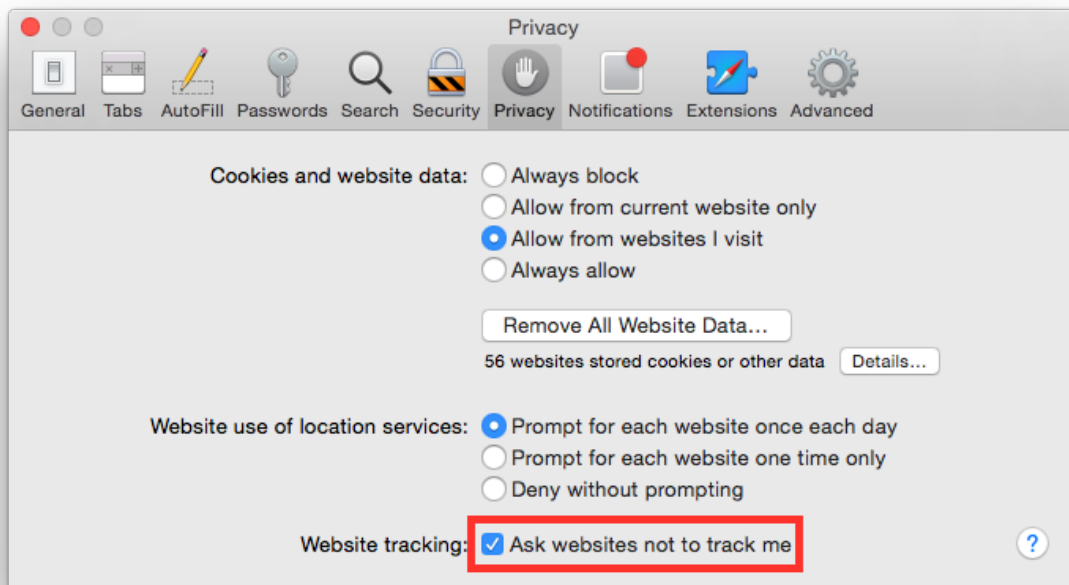


## Disable website tracking

Disables website tracking asking sites to do not track. Go to:

Open Safari Preferences Privacy

Check “Ask website not to track me”.



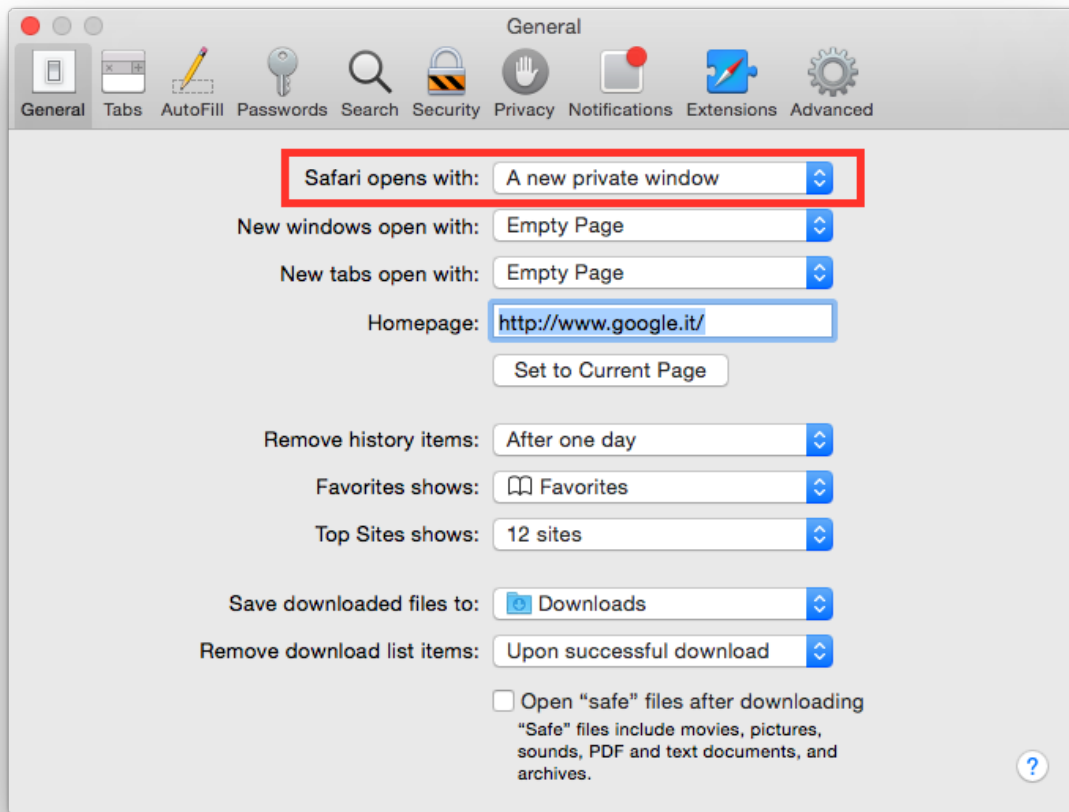


## Open with a private window

Set Safari to open only in new private window to keep your privacy. Go to:

Open Safari Preferences General

Set “Safari opens with” to “A new private window”.

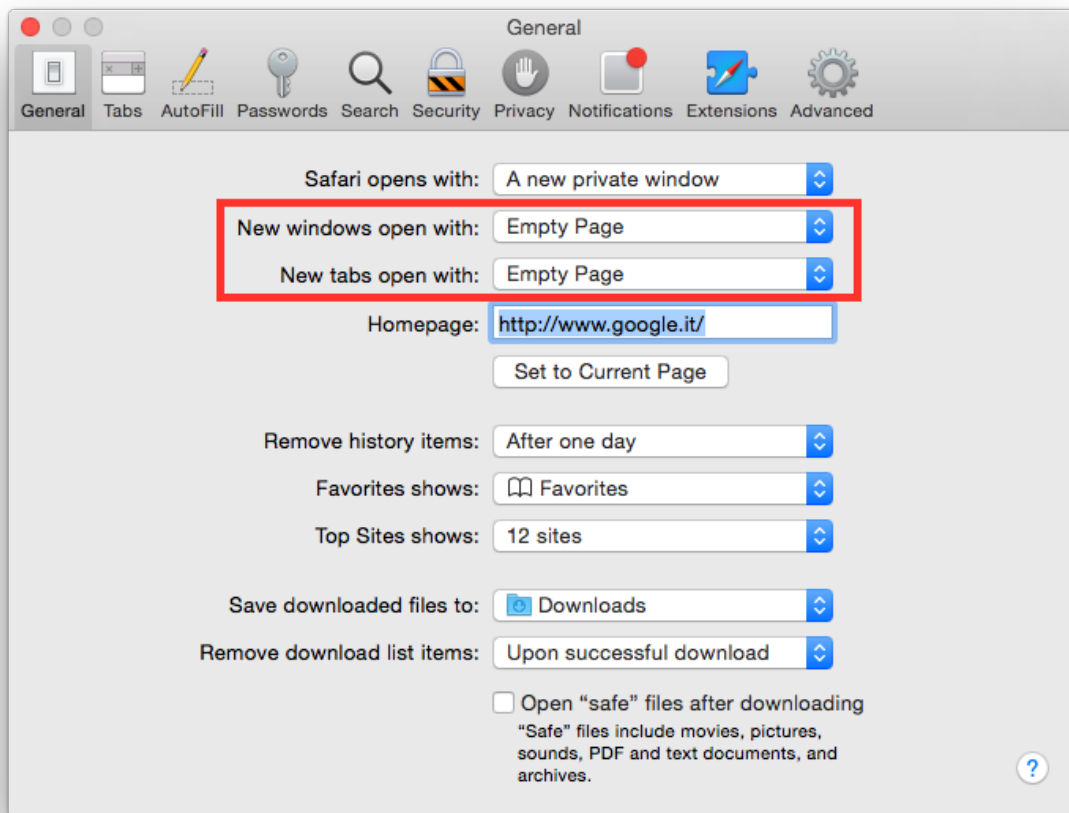


## Open with an empty page

Set Safari to open new windows and tabs with an empty page. Go to:

Open Safari Preferences General

Set “New windows open with” to “Empty Page”. Set “New tabs open with” to “Empty Page”.

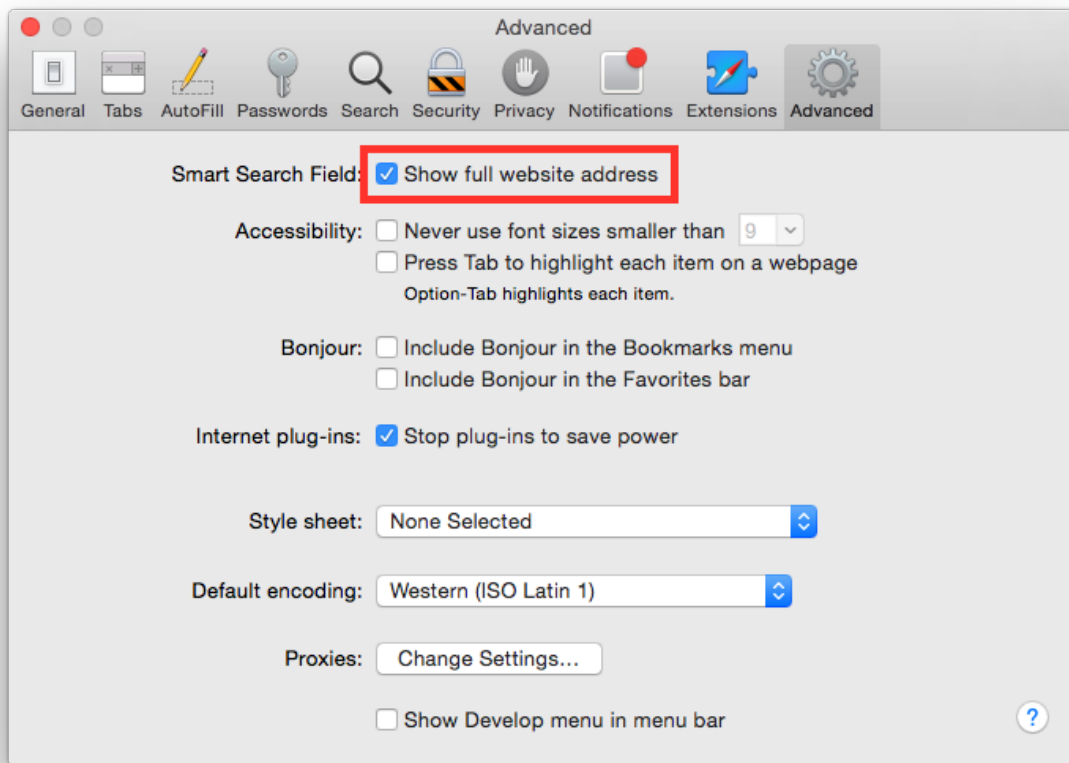


### Show website address

By default Safari shows only the domain in the address bar, it is suggested to show the whole website address. Go to:

Open Safari Preferences Advances

Check "Show full website address".

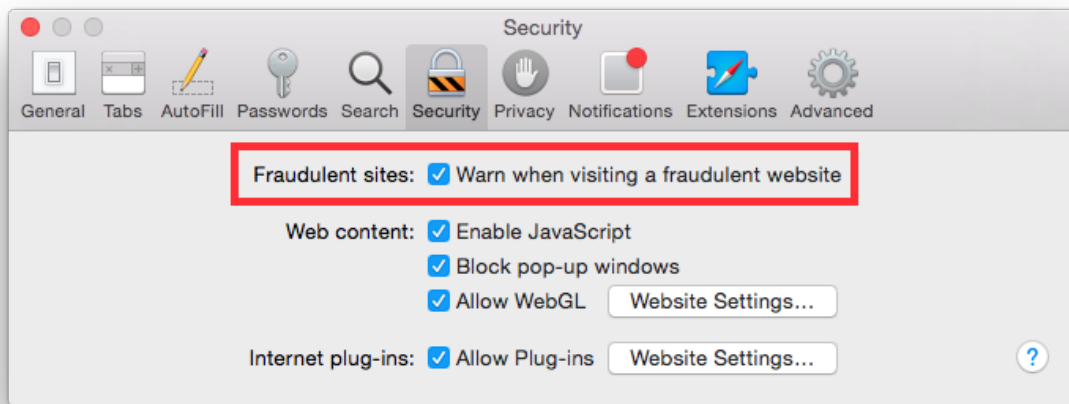


### Warn when visiting a fraudulent website

Safari can check the website you are visiting against a public and free feed on malicious websites. It is suggested to enable fraudulent website detection, go to:

Open Safari Preferences Security

Check “Warn when visiting a fraudulent website”.



### 1.1.4 Apple Safari 9

According to [Wikipedia](#) Safari is “a web browser developed by Apple Inc. included with the OS X and iOS operating systems. First released as a public beta on January 7, 2003, on the company’s OS X operating system, it became Apple’s default browser beginning with Mac OS X v10.3 “Panther”. The native browser of iOS is also called Safari, but has a different UI and uses a different WebKit version and API”.

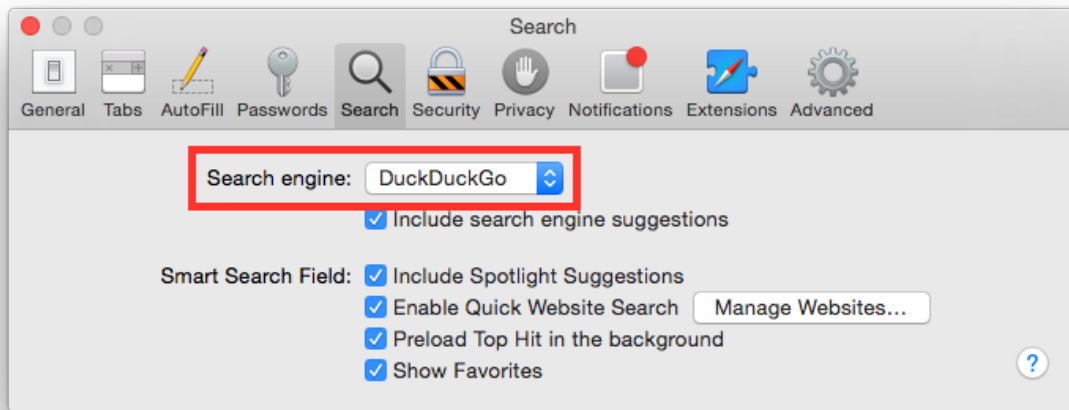
This chapter is dedicated to configuring Apple Safari version 9.x. It comes by default with Mac OS X 10.11 (El Capitan).

- *Change default search engine*
- *Clear history*
- *Control third party plugins*
- *Disable AutoFill*
- *Disable AutoFill username and passwords*
- *Disable open files after download*
- *Disable search suggestions*
- *Disable website tracking*
- *Open with a private window*
- *Open with an empty page*
- *Show website address*
- *Warn when visiting a fraudulent website*

#### Change default search engine

Change default search engine to [DuckDuckGo](#), it is a search engine who takes care of your privacy. Go to:

Open Safari Preferences Search  
Set “Search engine” to “DuckDuckGo”.

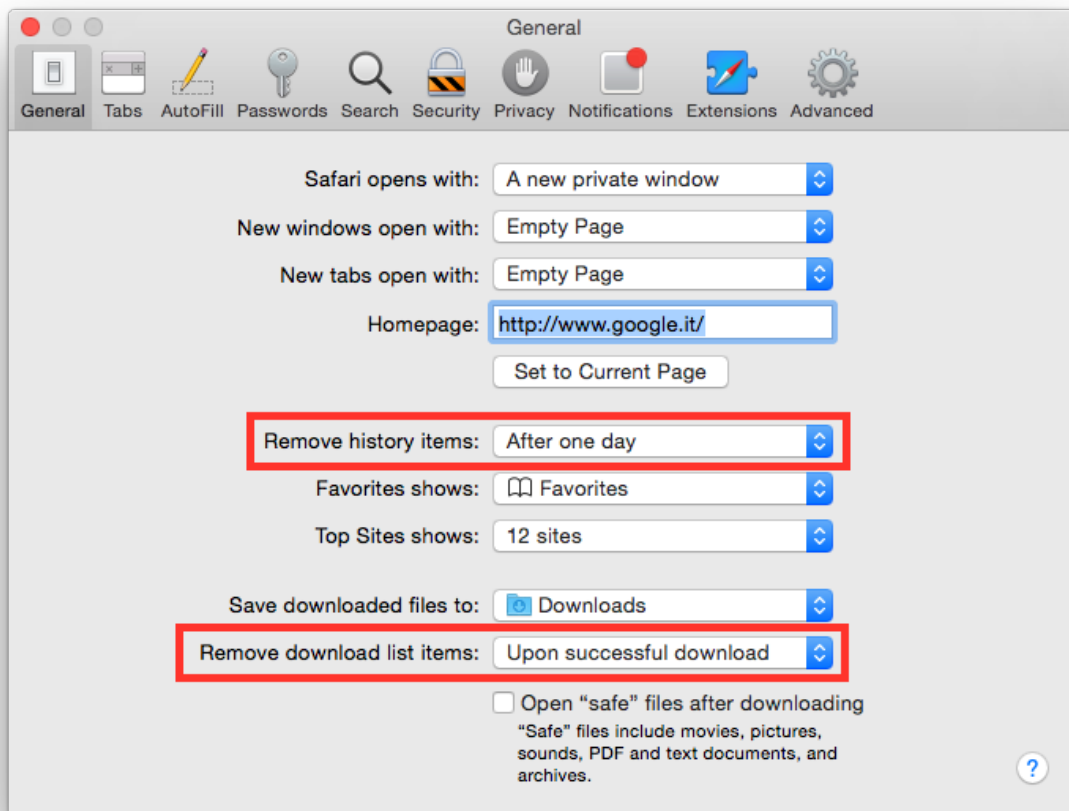


## Clear history

Used to never leave navigation traces in browser history. Go to:

Open Safari Preferences General

Set “Remove history items” to “After one day”. Set “Remove download list items” to “Upon successful download”.

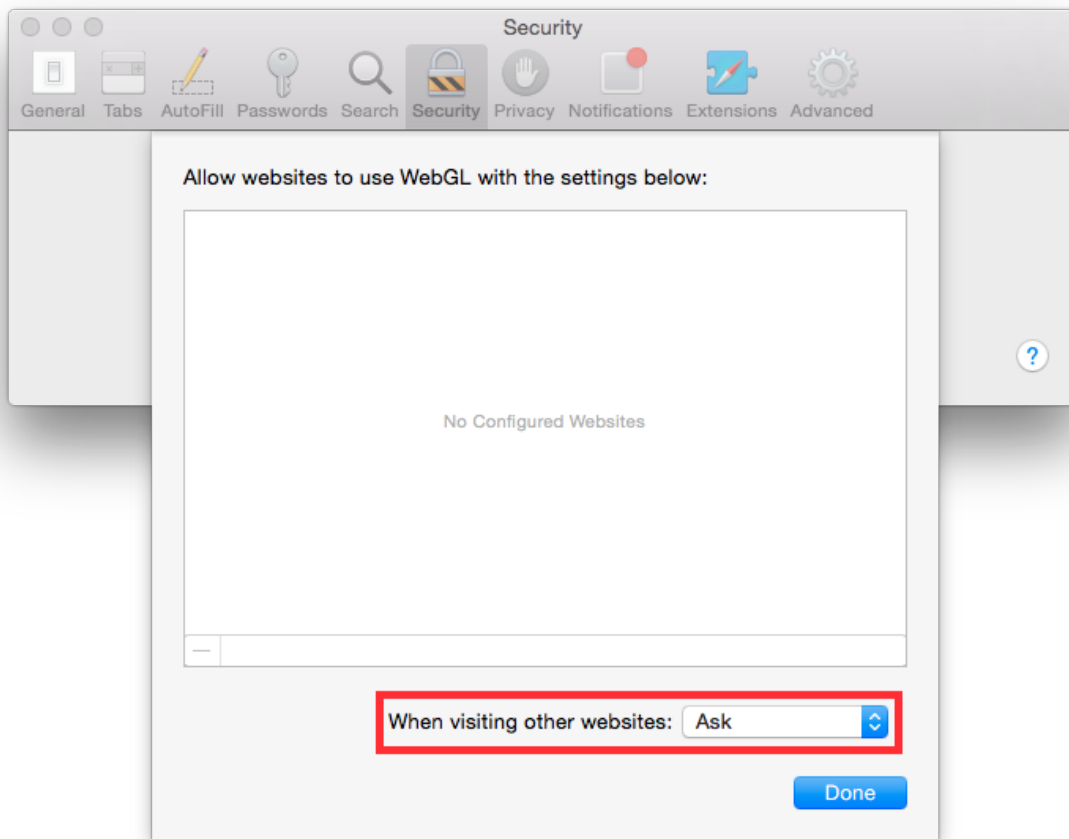


### Control third party plugins

Most browsers allow the continuous running of all third party scripts, giving malware an huge surface area of attack to get into your machine. Safari can ask for user permission each time a plugin is run, this is a good practice to control which website is asking your browser to run a plugin. It is suggested to control the run of WebGL plugins, go to:

Open Safari Preferences Security

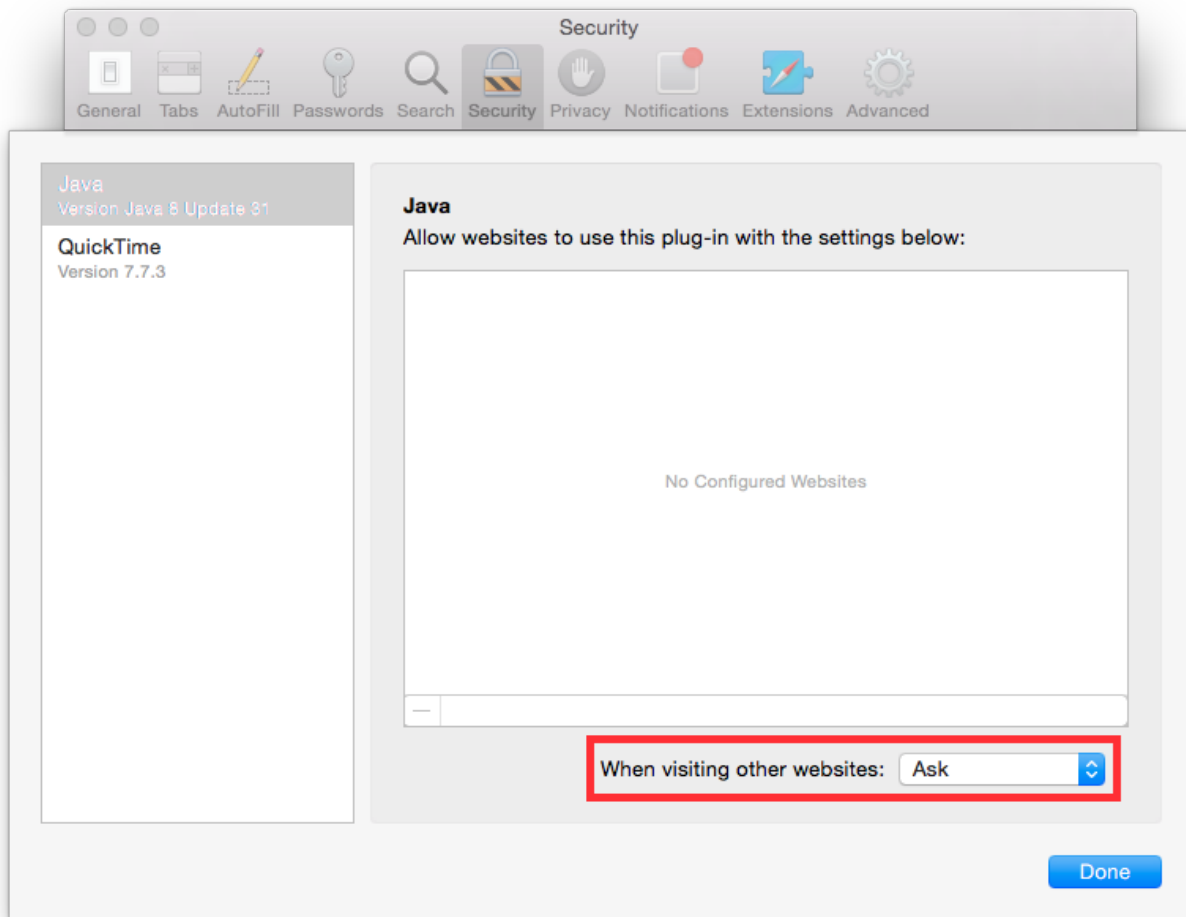
Check "Allow WebGL" and click on "Website Settings...", set "When visiting other websites" to "Ask".



It is suggested to control the run of WebGL plugins, go to:

Open Safari Preferences Security

Check "Allow Plug-ins" and click on "Website Settings...", set "When visiting other websites" to "Ask".



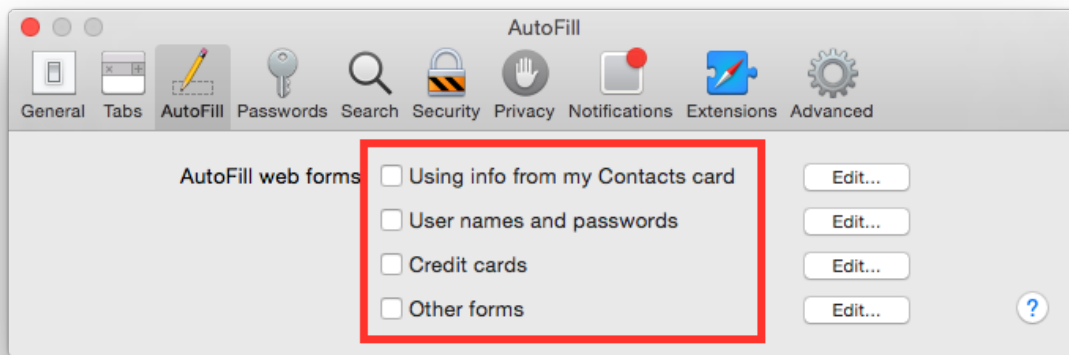
## Disable AutoFill

Disables automatic fill of forms. Go to:

Open Safari Preferences AutoFill

Uncheck all boxes.



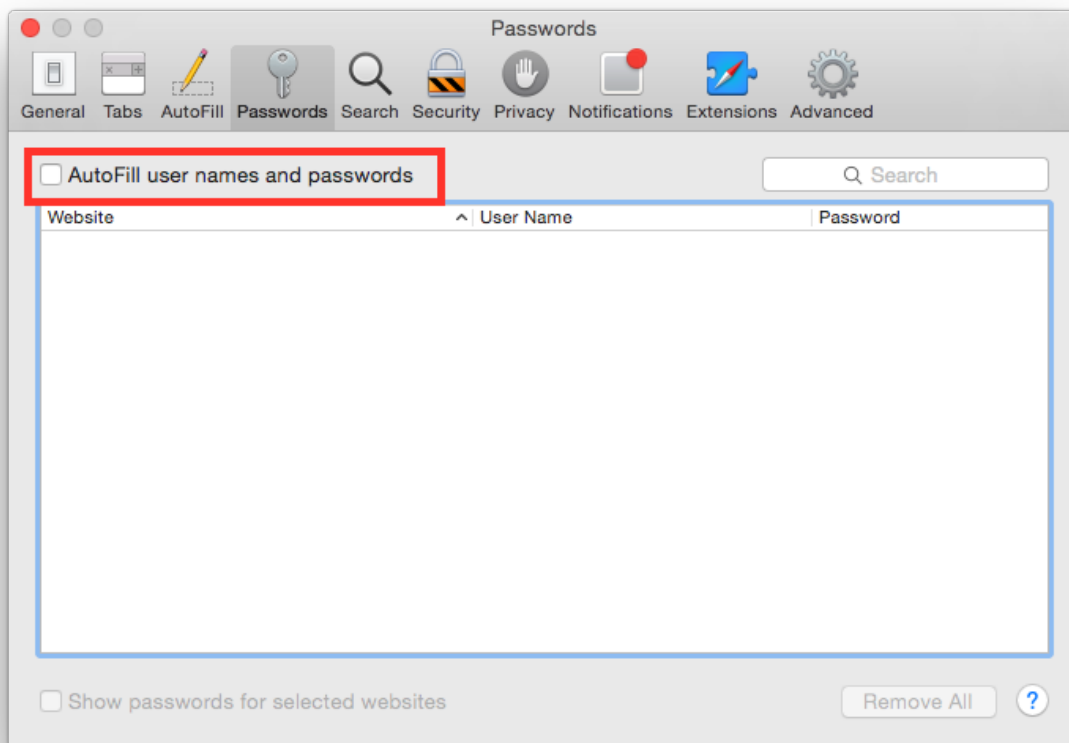


### Disable AutoFill username and passwords

Disables automatic fill of forms with usernames and passwords. Go to:

Open Safari Preferences Passwords

Uncheck "AutoFill user names and passwords".

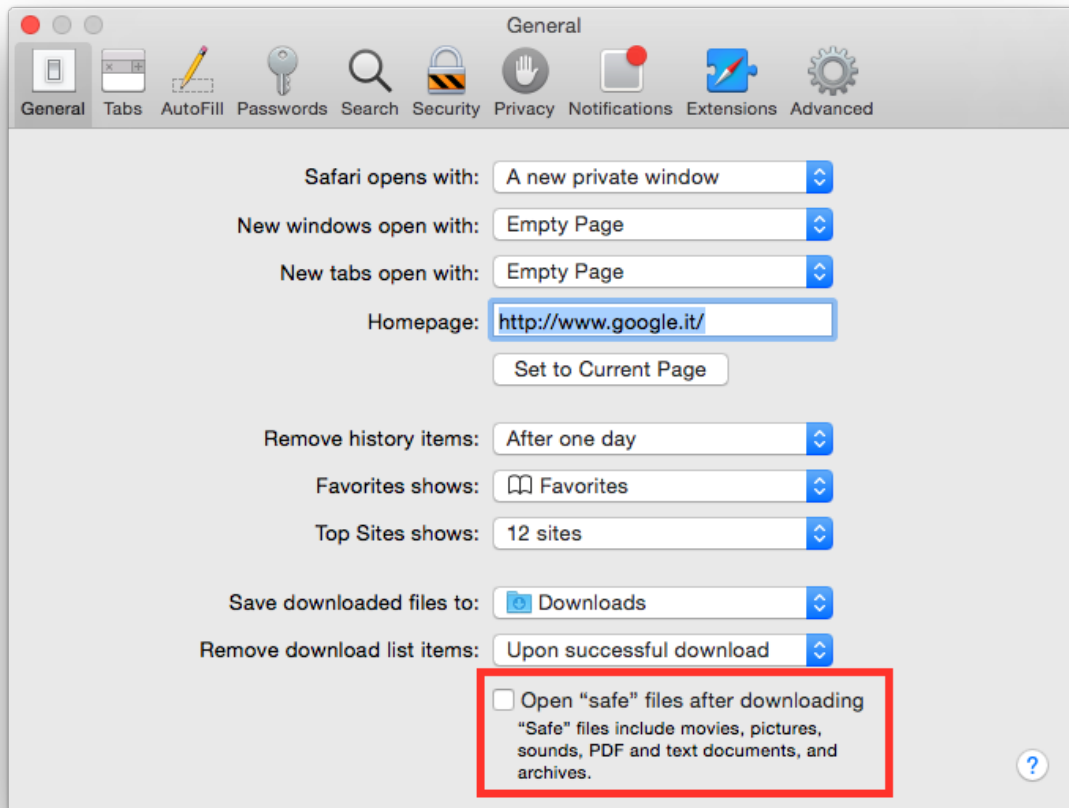


### Disable open files after download

Disables automatic opening of downloaded file, even if they are safe. It is suggested to never run arbitrary files downloaded. Go to:

Open Safari Preferences General

Uncheck “Open ‘safe’ files after downloading”.

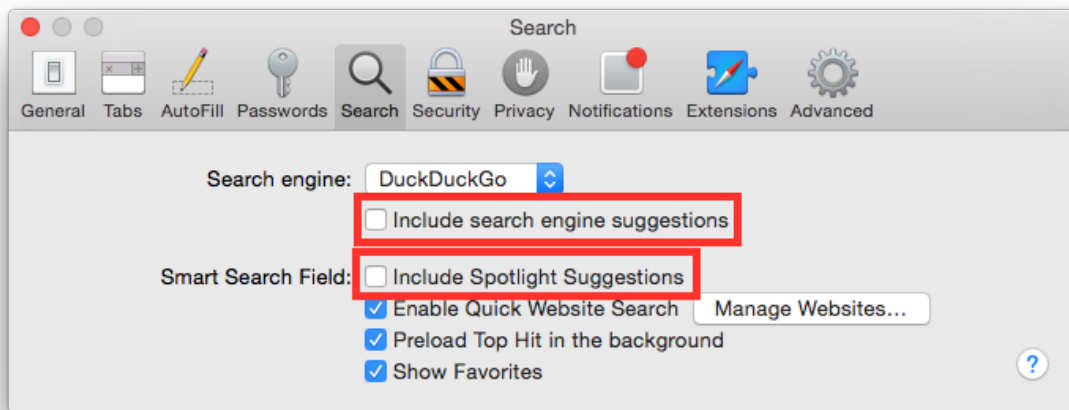


### Disable search suggestions

Disables suggestions to avoid leaking potential data when typing in the search box. Go to:

Open Safari Preferences Search

Uncheck “Include search engine suggestions”. Uncheck “Include Spotlight suggestions”.

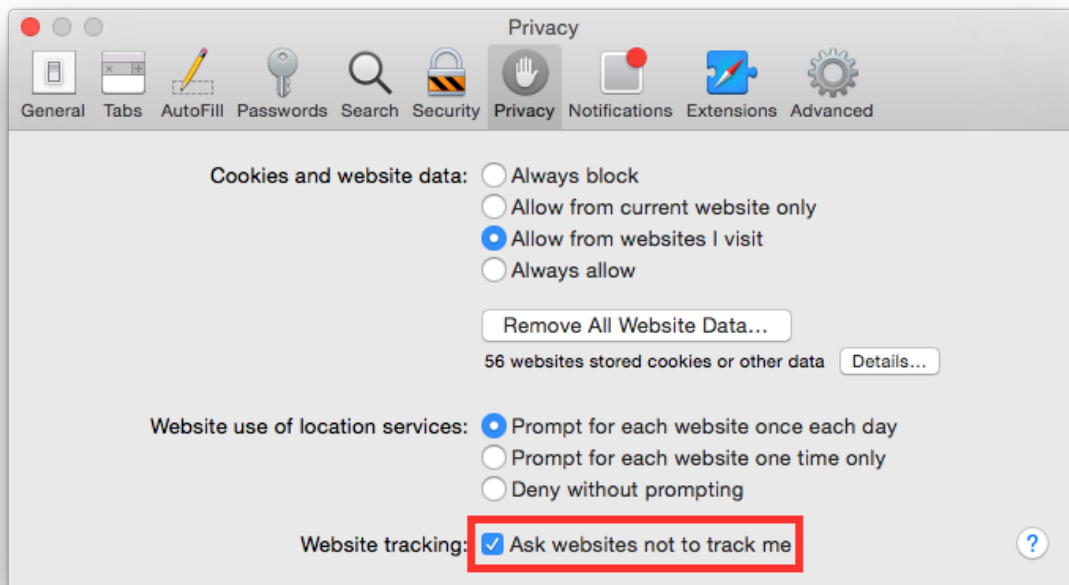


### Disable website tracking

Disables website tracking asking sites to do not track. Go to:

Open Safari Preferences Privacy

Check “Ask website not to track me”.

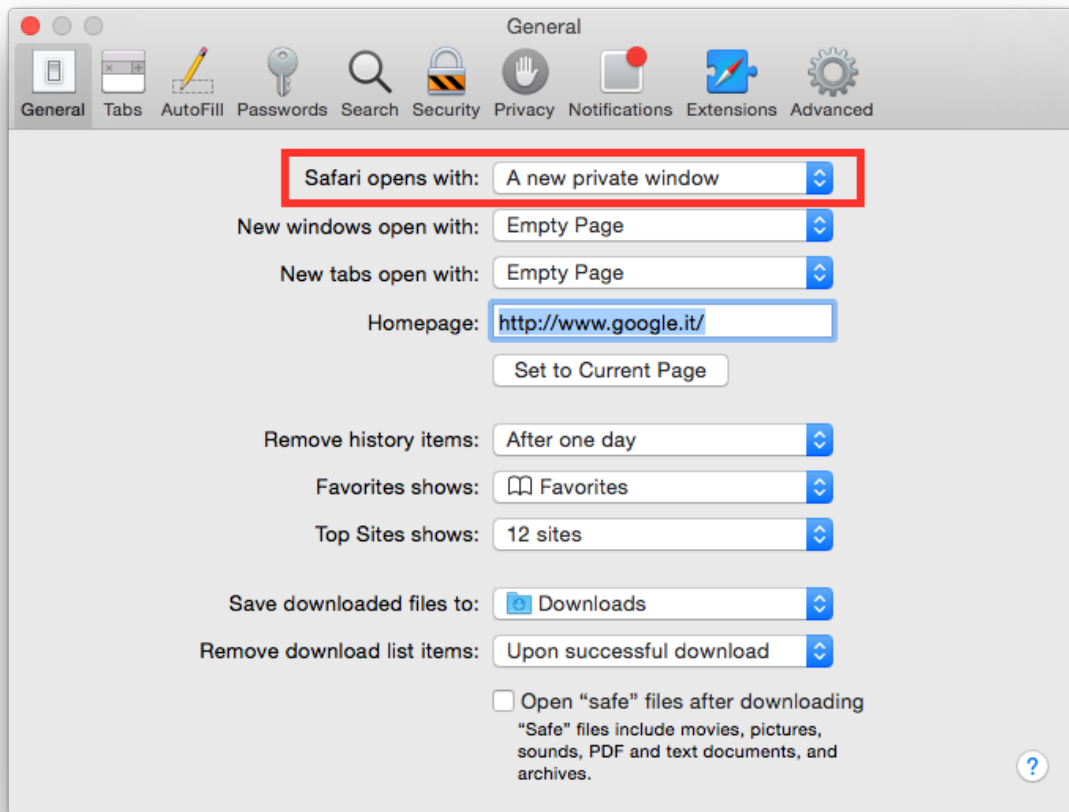


### Open with a private window

Set Safari to open only in new private window to keep your privacy. Go to:

Open Safari Preferences General

Set “Safari opens with” to “A new private window”.

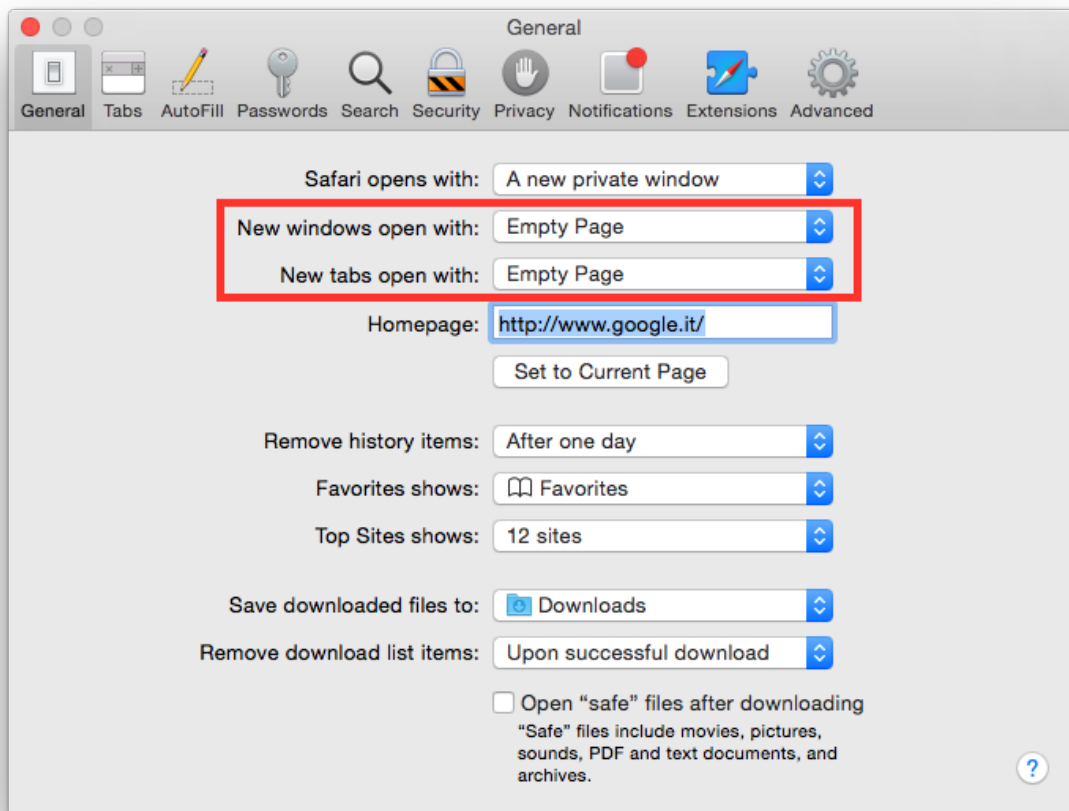


### Open with an empty page

Set Safari to open new windows and tabs with an empty page. Go to:

Open Safari Preferences General

Set “New windows open with” to “Empty Page”. Set “New tabs open with” to “Empty Page”.

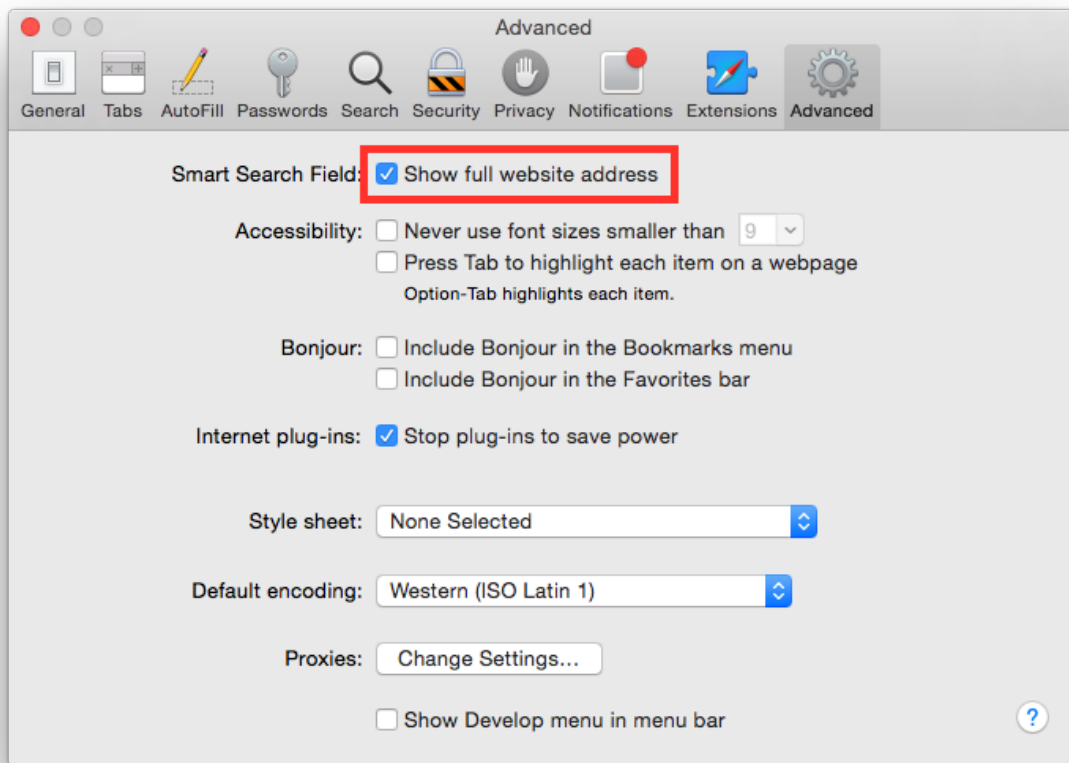


### Show website address

By default Safari shows only the domain in the address bar, it is suggested to show the whole website address. Go to:

Open Safari Preferences Advanced

Check "Show full website address".

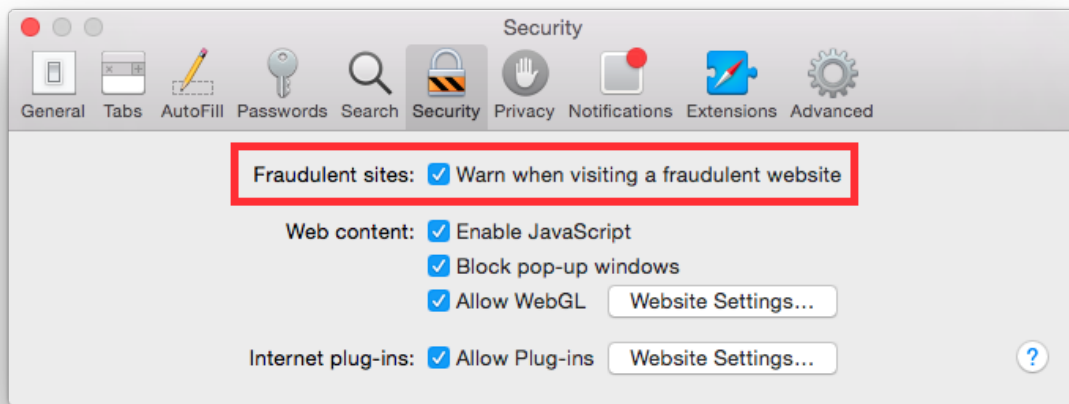


### Warn when visiting a fraudulent website

Safari can check the website you are visiting against a public and free feed on malicious websites. It is suggested to enable fraudulent website detection, go to:

Open Safari Preferences Security

Check “Warn when visiting a fraudulent website”.



### 1.1.5 GPG Suite

According to [GPG Tools official website](#) GPG Tools is used “to encrypt, decrypt, sign and verify files or messages. Manage your GPG Keychain with a few simple clicks and experience the full power of GPG”. GPG Suite is an implementation of GPG for Mac OS X with a keychain and an Apple Mail extension.

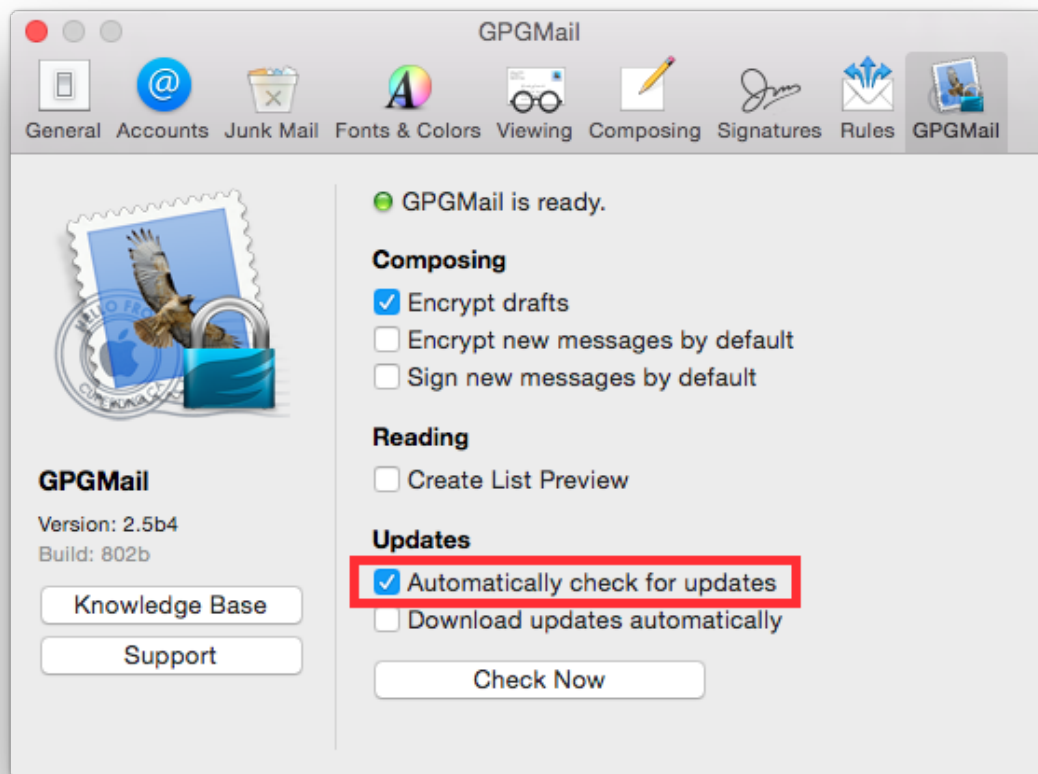
- *Automatic update check*
- *Check installation package signature*
- *Encrypt drafts*

#### Automatic update check

It is recommended to keep software always updated. To enable automatic update check, go to:

Open Apple Mail Preferences GPGMail panel

Check “Automatically check for updates” option.



### Check installation package signature

The installation package is provided with an hash signature and a GPG signature. It is recommended to check digital signature before the installation.

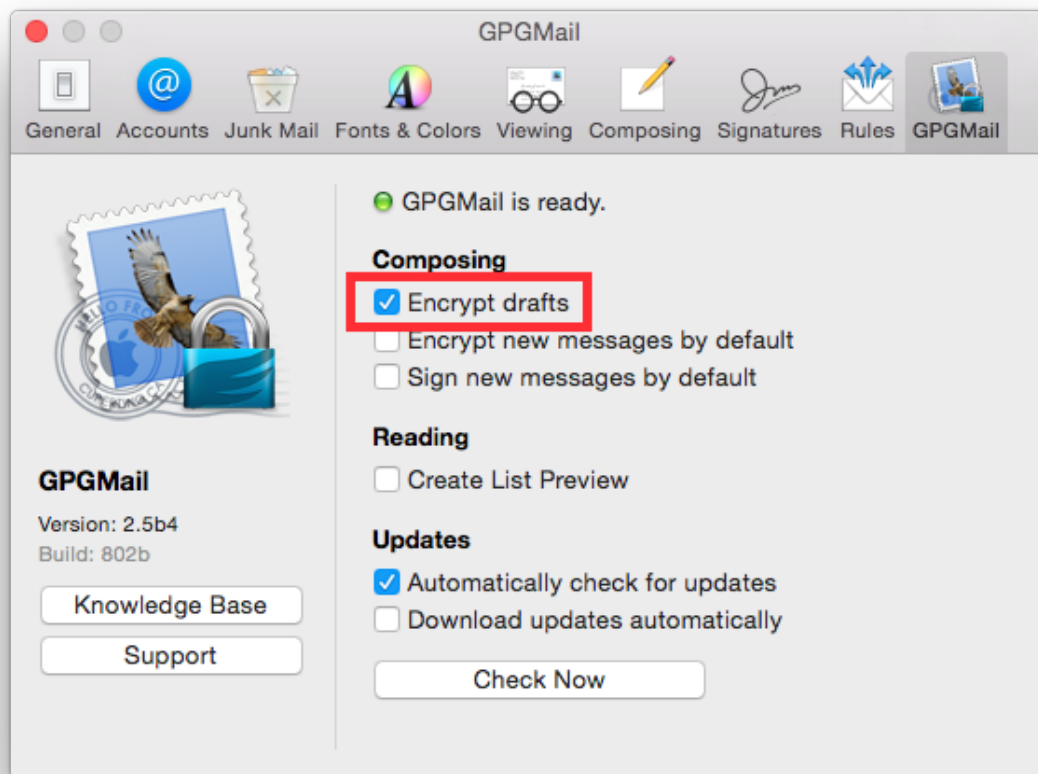
### Encrypt drafts

It is recommended to store mail drafts in an encrypted format, to avoid leak of draft emails saved in clear text. To enable drafts encryption, go to:

Open Apple Mail Preferences GPGMail panel

Check “Encrypt drafts” option.





### 1.1.6 LittleSnitch 3

According to the vendor [website](#) LittleSnitch is “a firewall that intercepts unwanted connection attempts, and lets you decide how to proceed”.

It is really common, and a best practice, to replace Mac OS X default firewall with a more advanced firewall like Little Snitch.

This chapter is dedicated to configuring Little Snitch 3.

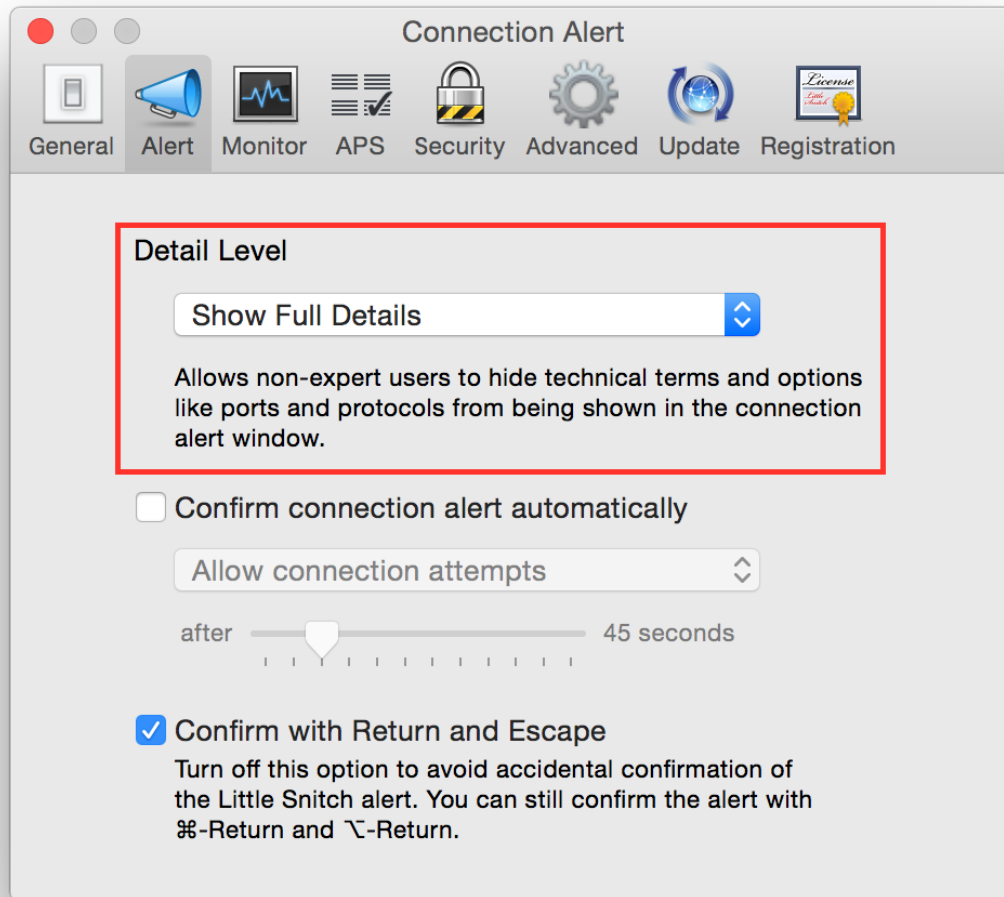
- *Show full details*
- *Enable update check*
- *Disable scripting access*
- *Mark ruled as unapproved*

#### Show full details

It is suggested to enable an advanced mode to all event’s details. Start Little Snitch. Go to:

Preferences Alert

Set “Detail Level” to “Show Full Details”.

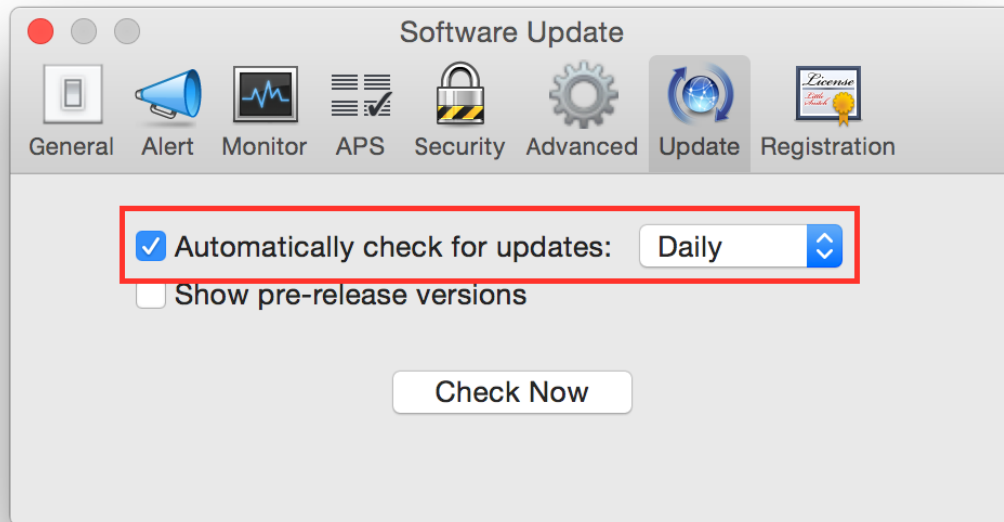


### Enable update check

By default automatic update check is disabled, it is suggested to enable it. Start Little Snitch. Go to:

Preferences Update

Check “Automatic check for updates” and set it to “Daily”.

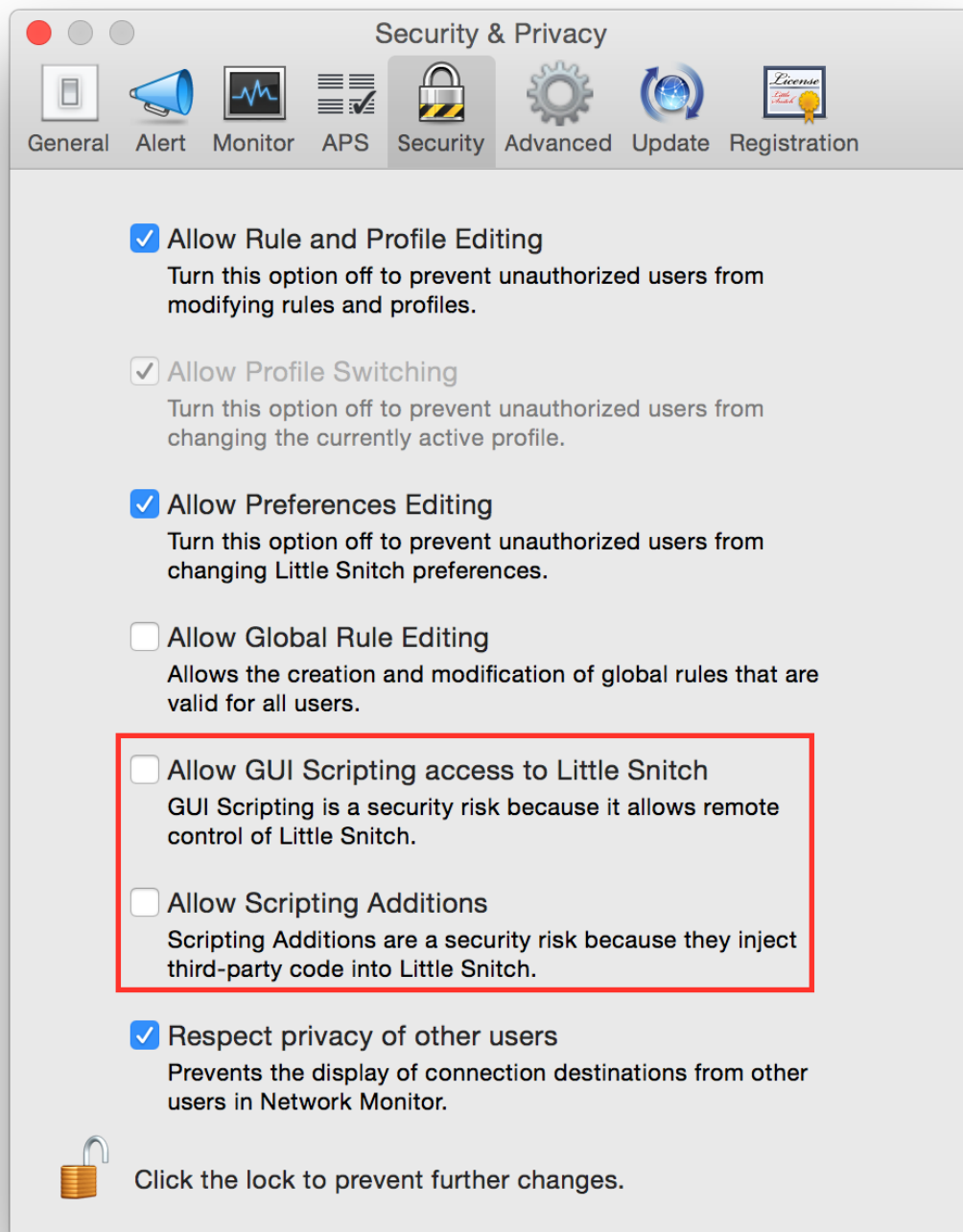


### Disable scripting access

Scripting access is a security risk because a malicious software could be able to add rules to LittleSnitch and/or modify its behavior. It is suggested to disable it, it is usually disabled by default. Start Little Snitch. Go to:

Preferences Security

Un-check “Allow GUI Scripting access to Little Snitch”. Un-check “Allow Scripting Additions”.



### Mark ruled as unapproved

By default rules created with the alert window are auto approved. It is a good practice to create it as not approved, and review them all later. Start Little Snitch. Go to:

Preferences Advanced

Check “Mark rules from connection alert as unapproved”.



### 1.1.7 MongoDB

According to [MongoDB official website](#) MongoDB is “a document database that provides high performance, high availability, and easy scalability”.

This chapter is dedicated to configuring MongoDB version 2.x.

- *Authentication*
- *Authorization*
- *Disable Localhost Exception*
- *Disable server side scripting*
- *Disable status interface*
- *Disable the REST interface*
- *Encryption*
- *Limit Network Exposure*
- *Run MongoDB with a dedicated user*
- *References*

### Authentication

Authentication is the process of verifying the identity of a client or a user. MongoDB supports different authentication mechanisms, it is suggested to always use authentication for all users and clients (with different credentials for each one). Even if you have deployed MongoDB servers in a trusted network it is good security practice to enable authentication. Please refer to MongoDB documentation to understand how create and use users over different authentication mechanisms.

### Authorization

Authorization is a set of roles to give users permissions that pair resources with allowed operations. It is suggested to use authorization to fine tune users profiles and let each user access the data or run the operations it needs. MongoDB does not enable authorization by default, you can enable authorization using the `--auth` option. Example:

```
$ mongod --auth
```

Or set it in the configuration file:

```
auth = true
```

Please refer to MongoDB documentation to understand how to work with authorization mechanisms.

### Disable Localhost Exception

The localhost exception allows you to enable authorization before creating the first user in the system. When active, the localhost exception allows all connections from the localhost interface to have full access to that instance. The exception applies only when there are no users created in the MongoDB instance. To prevent unauthorized access to a cluster's shards, you must either create an administrator on each shard or disable the localhost exception. To disable the localhost exception, add `setParameter` to set the `enableLocalhostAuthBypass` parameter to 0 during startup. Example:

```
$ mongod --setParameter enableLocalhostAuthBypass=0
```

Or set it in the configuration file:

```
setParameter = enableLocalhostAuthBypass=0
```

### Disable server side scripting

In some server-side operations (i.e. `mapReduce`, `group`, `eval`, `$where`), MongoDB supports the execution of JavaScript code. To mitigate the exploiting of a possible application level vulnerability, if you do not use these operations, it is suggested to disable server-side scripting. To disable server-side scripting add `noscripting` parameter during startup. Example:

```
$ mongod --noscripting
```

Or set it in the configuration file:

```
noscripting = false
```

## Disable status interface

The status interface is an HTTP server exposing a web page that contains some statistics that may of interest to system administrators. It is suggested to disable the status interface to not expose an unused service. To disable the status interface add *nohttpinterface* argument during startup. Example:

```
$ mongod --nohttpinterface
```

Or set it in the configuration file:

```
nohttpinterface = true
```

Since version 2.6 MongoDB disables the HTTP interface by default.

## Disable the REST interface

The REST interface is a fully interactive administrative REST interface, which is disabled by default. This interface does not support any authentication and you should always restrict access to this interface to only allow trusted clients to connect to this port. It is suggested to leave this interface disabled, removing the following arguments by the command line, if present:

```
$ mongod --rest --httpinterface
```

Or disable it in the configuration file:

```
rest = false
```

If you have to leave this interface enabled, you should only allow trusted clients to access this service (using proper firewall rules).

## Encryption

MongoDB clients can use SSL to encrypt connections to mongo instances. It is suggested to always use SSL encryption when accessing MongoDB over a network.

Please refer to MongoDB documentation to understand how to setup SSL encryption.

## Limit Network Exposure

Restriction access to the database service is a critical aspect of service security. It is suggested to do not expose your database to resources that are not in need to access it. You can use the *-bind\_ip* option on the command line at run time or the *bindIp* in the configuration file to limit the network accessibility of a MongoDB program. Example:

```
$ mongod --bind_ip 127.0.0.1
```

Or set it in the configuration file:

```
bind_ip = 127.0.0.1
```

If you need fine tuned network access limitation not limited to binding on an interface is suggested to use a firewall to place custom network traffic ACLs.

### Run MongoDB with a dedicated user

Privilege separation should always be used, it is suggested to run MongoDB processes with a dedicated user account (an operative system account with the minimum privileges needed to run the service). Most installers already creates a dedicated user when installing MongoDB.

### References

- <http://docs.mongodb.org/v2.6/MongoDB-security-guide.pdf>

### 1.1.8 MySQL Server

According to [MySQL official website](#) MySQL is “*open-source relational database management system (RDBMS)*”.

- *Connection Encryption*
- *Connection Error Limit*
- *Disable LOAD DATA LOCAL INFILE*
- *Disable SHOW DATABASES*
- *Hardening Script*
- *Interface Binding*
- *Privilege Hardening*
- *Rename root User*
- *References*

### Connection Encryption

By default MySQL connections are not encrypted and everything flows over network in open text. If you are using MySQL over a network it is suggested to use encryption, refer to [MySQL documentation](#) to understand how to configure an encryption mechanism.

### Connection Error Limit

It is suggested to apply host ban to clients with many unsuccessful authentications. As stated in [MySQL documentation](#):

*If there are more than this number of interrupted connections from a host, that host is blocked from further connections. You can unblock blocked hosts with the FLUSH HOSTS statement. If a connection is established successfully within fewer than max\_connect\_errors attempts after a previous connection was interrupted, the error count for the host is cleared to zero. However, once a host is blocked, the FLUSH HOSTS statement is the only way to unblock it.*

Edit the configuration file *my.cnf* and set *max\_connect\_errors*:

```
max_connect_errors = 3
```



## Disable LOAD DATA LOCAL INFILE

The LOAD DATA LOCAL INFILE command allows users, or an attacker, to read local files and even access other files on the operating system. It is also a common command used by attackers exploiting by methods such as SQL injection. It is suggested to disable the command, edit the configuration file *my.cnf* and set *local-infile*:

```
local-infile=0
```

## Disable SHOW DATABASES

SHOW DATABASES is a command used by users, or attackers, to list all databases available. Stripping remote attackers of their information gathering capabilities is critical to a secure security posture. It is suggested to disable the command, edit the configuration file *my.cnf* and add *skip-show-database* to the [mysqld] section

```
[mysqld]  
skip-show-database
```

## Hardening Script

MySQL comes with an hardening script to check database server security and remove some default settings. You can run it with the command:

```
mysql_secure_installation
```

It will ask you for your desired hardening level through some questions.

## Interface Binding

If you don't need to access your database from another machine it is suggested to bind MySQL service on localhost only, edit the configuration file *my.cnf* and set *bind-address*:

```
bind-address = 127.0.0.1
```

You can also disable networking if not used with *skip-networking* option.

## Privilege Hardening

You should carefully manager users and privileges, it is suggested to follow at least these best practices:

- Each application that uses MySQL should have its own user that only has limited privileges and only has access to the databases it needs to run.
- Never use ALL TO ..
- Never use % for a hostname
- Application user permissions should be restrictive as possible
- Only allow super privileges to dba accounts, and localhost
- Never ever give users global privileges, except for root, backup user, monitoring user, replication user
- Take extra caution when granting SUPER or FILE privileges: SUPER can modify runtime configuration and become other users, FILE allows reading or writing files as MySQL process

### Rename root User

It is suggested to change the root login name. If an attacker is trying to access the root MySQL login, they will need to perform the additional step of finding the username.

The root login can be changed with the following SQL commands:

```
RENAME USER 'root'@'localhost' TO 'foobar'@'localhost';  
FLUSH PRIVILEGES;
```

### References

- [https://www.owasp.org/index.php/OWASP\\_Backend\\_Security\\_Project\\_MySQL\\_Hardening](https://www.owasp.org/index.php/OWASP_Backend_Security_Project_MySQL_Hardening)

### 1.1.9 Nginx

According to [Nginx official website](#) Nginx is “*is an HTTP and reverse proxy server, a mail proxy server, and a generic TCP proxy server, originally written by Igor Sysoev. For a long time, it has been running on many heavily loaded Russian sites including Yandex, Mail.Ru, VK, and Rambler.*”.

- *Catch all deny virtualhost*
- *Enable Anti-Clickjacking Header*
- *Enable HTTP Strict Transport Security*
- *Enable X-XSS Protection*
- *Deny access to some resources*
- *Deny illegal Host headers*
- *Disable Autoindex*
- *Disable Content-type Sniffing*
- *Disable Server Signature*
- *Disable SSI module*

### Catch all deny virtualhost

A catch all virtual host is the website server when your website is accessed by IP address and not by hostname. It is usually used only by bots and attackers, so it is suggested to setup a virtual host listening on your IP website and deny all requests. Create a virtualhost with the following configuration:

```
server {  
    listen 80 default;  
    server_name _;  
    deny all;  
}
```

## Enable Anti-Clickjacking Header

The X-Frame-Options will instruct a browser to load the resources only from the same origin, this means the page can't load inside a framed tag (i.e. frame or iframe). Use this only if your business doesn't plan to have the site loaded in a frame. Add the following code to your virtual host or server block of your site:

```
add_header X-Frame-Options "SAMEORIGIN";
```

## Enable HTTP Strict Transport Security

If it apply to your website, it is suggested to consider enabling HSTS (HTTP Strict Transport Security) mechanism, which let browsers to communicate with your websites only over HTTPS protocol. This mechanism is designed to reduce man in the middle attacks (MiTM). In order to enable HSTS on Nginx, you should need to add this code to your virtual host or server block of your site:

```
add_header Strict-Transport-Security max-age=15768000;
```

Example of a redirect virtual host with HSTS enabled:

```
server {
    listen 80;
    add_header Strict-Transport-Security max-age=15768000;
    return 301 https://www.example.com$request_uri;
}
```

## Enable X-XSS Protection

The X-XSS protection is used to mitigate Cross-Site scripting attacks. Add the following code to your virtual host or server block of your site:

```
add_header X-XSS-Protection "1; mode=block";
```

## Deny access to some resources

Sometimes critical data are published during application deploy. It is suggested to deny access to sensitive resources, for example the *.git* folder, adding a location statement to deny access to *.git* or another resource:

```
server {
    location ~ /\.git {
        deny all;
    }
}
```

## Deny illegal Host headers

Malicious bots or vulnerability probing usually sends also requests with an improper or empty Host header. The default technique to block this kind of attempts is to use a "Catch all virtualhost", but in some cases, for example if your website is SSL/TLS encrypted, you can't use a default virtualhost. It is suggested to block all requests with an illegal Host header with the following configuration (example.com is your website in this example):

```
server {
    # Deny illegal Host headers.
    if ($host !~* ^(example.com|www.example.com)$ ) {
        return 444;
    }
}
```

The returned HTTP error code 444 is used in Nginx logs to indicate that the server has returned no response to the client and closed the connection (useful to block malicious requests).

### Disable Autoindex

It is suggested to disable the autoindex module, disabling the directive *autoindex* in your command location block:

```
autoindex off;
```

### Disable Content-type Sniffing

This header will prevent the browser from interpreting files as something else than declared by the content type in the HTTP headers. Add the following code to your virtual host or server block of your site:

```
add_header X-Content-Type-Options nosniff;
```

### Disable Server Signature

By default Nginx sends banner with version number, it is suggested to disable server banner disabling *server\_tokens* in global configuration file:

```
server_tokens off;
```

For more information see: [http://wiki.nginx.org/HttpCoreModule#server\\_tokens](http://wiki.nginx.org/HttpCoreModule#server_tokens)

### Disable SSI module

It is suggested to disable the HTTP SSI module. Add the following code to your virtual host or server block of your site:

```
ssi off;
```

For more information see: [http://nginx.org/en/docs/http/nginx\\_http\\_ssi\\_module.html](http://nginx.org/en/docs/http/nginx_http_ssi_module.html)

## 1.1.10 OpenSSH

According to [OpenSSH official website](#) OpenSSH is used “*OpenSSH is a free version of the SSH connectivity tools that technical users of the Internet rely on. Users of telnet, rlogin, and ftp may not realize that their password is transmitted across the Internet unencrypted, but it is. OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other attacks. Additionally, OpenSSH provides secure tunneling capabilities and several authentication methods, and supports all SSH protocol versions*”. Different versions of OpenSSH support different options which are not always compatible. This guide show settings for the most commonly deployed OpenSSH versions.

- *Change port number*
- *Compression after authentication*
- *Configure Idle Log Out Timeout Interval*
- *Enable strict mode*
- *Enable a Warning Banner*
- *Disable .rhosts Files*
- *Disable Challenge Response*
- *Disable Empty Passwords*
- *Disable gateway for forwarded ports*
- *Disable Host-Based Authentication*
- *Disable Password Authentication*
- *Disable Protocol 1*
- *Disable Roaming*
- *Disable Root Logins*
- *Disable SSH forwarding*
- *Disable TCP forwarding*
- *Disable user environment*
- *Disable X11 forwarding*
- *Display a warning message before login*
- *Do not use SSH Agent Forwarding*
- *Hash Known Hosts*
- *Key storage*
- *Increase Key Strength*
- *Limit port forwarding*
- *Limiting brute forcing attempts*
- *Message authentication codes*
- *OTP Setup*
- *Restrict IP Listen Address*
- *Reduce Grace Time*
- *Route traffic over TOR*
- *Symmetric ciphers*
- *Use PAM*
- *Use privilege separation*
- *Use strong key algorithms*
- *Whitelisting / blacklisting users*

- *Whitelisting / blacklisting groups*
- *References*

### Change port number

SSH default port (22/tcp) is a service target of worms, script kiddies, and all kind of brute forcing around. It is suggested to edit `sshd_config` file (usually located in `/etc/ssh/sshd_config`) to run the SSH daemon on a non default port, using the *Port* option:

```
Port 34567
```

### Compression after authentication

It is suggested to enable compression only after authentication. Open `sshd_config` (usually located in `/etc/ssh/sshd_config`) and make sure following value is configured:

```
Compression delayed
```

### Configure Idle Log Out Timeout Interval

Users can login to server via ssh, it is suggested to set an idle timeout interval to avoid unattended ssh session. Open `sshd_config` (usually located in `/etc/ssh/sshd_config`) and make sure following values are configured:

```
ClientAliveInterval 300
ClientAliveCountMax 0
```

### Enable strict mode

Using strict mode you can enforce some checks on important files inside users' home directory have the proper privileges and ownership, SSH daemon will only allow a remote user to log on if checks pass. It is suggested to enable strict mode editing `sshd_config` file and enabling *StrictModes*:

```
StrictModes yes
```

### Enable a Warning Banner

Set a warning banner by updating `sshd_config` with the following line:

```
Banner /etc/issue
```

This setting is suggested *only* on intranet facing servers. If you are using a custom banner on an internet facing system you are disclosing some kind of information and it is quite easy to fingerprint and track your system. For example think about your “fingerprint prone” SSH server published as an hidden node. Anyone could correlate the unique banner with you.

## Disable .rhosts Files

SSH can be configured to emulate the behavior of the obsolete rsh command honoring *.rhosts* files. This is historically unsafe and it is suggested to disable it, edit *sshd\_config* file and disable *IgnoreRhosts*:

```
IgnoreRhosts yes
```

## Disable Challenge Response

You should also disable challenge-response authentication, in case your version of OpenSSH is using PAM to authenticate. It is suggested to edit *sshd\_config* file and disable *ChallengeResponseAuthentication*:

```
ChallengeResponseAuthentication no
```

## Disable Empty Passwords

You need to explicitly disallow remote login from accounts with empty passwords, update *sshd\_config* with the following line:

```
PermitEmptyPasswords no
```

## Disable gateway for forwarded ports

SSH binds local port forwardings to the loopback address only, as default. This is a security feature to prevent other remote hosts from connecting to forwarded ports. The *GatewayPorts* option can be used to specify if this is the expected behaviour. It is suggested to disable *GatewayPorts*, it is already disabled by default in most distributions, edit *sshd\_config* file and disable *IgnoreRhosts*:

```
GatewayPorts no
```

## Disable Host-Based Authentication

It is suggested to disable host-based authentication, as *.rhost* based authentication, it is not rock solid authentication. To disable host-based authentication, edit *sshd\_config* file and disable *HostbasedAuthentication*:

```
HostbasedAuthentication no
```

## Disable Password Authentication

By default SSH can use keys or password to provide authentication, passwords are prone to brute force attacks. It is suggested to use keys only and completely disable password-based logins. To stop password based authentication, edit *sshd\_config* file and disable *PasswordAuthentication*:

```
PasswordAuthentication no
```

### Disable Protocol 1

The legacy SSH protocol 1 is not secure: it suffers of man-in-the-middle attacks and it has a myriad of vulnerabilities; it should be disabled although in most cases it already is by default. It is suggested to edit *sshd\_config* file and add the following line to use only SSH protocol version 2:

```
Protocol 2
```

### Disable Roaming

OpenSSH has some undocumented, and rarely used features. It is suggested to disable roaming feature, in the past it leads to a known vulnerability. Add to *ssh\_config* file:

```
Host *  
    UseRoaming no
```

### Disable Root Logins

It is suggested to not enable root login via SSH, this account has high privileges and it is usually target of attacks. A good practice is to login with a normal user, the root account is still available by using *su* and *sudo* tools. To disallow logins with user root, edit */sshd\_config* file and make sure you have the following entry:

```
PermitRootLogin no
```

### Disable SSH forwarding

Port forwarding via SSH (SSH tunneling) creates a secure connection between a local computer and a remote machine through which services can be relayed. It is suggested to disable this feature, update *sshd\_config* with the following line:

```
AllowTcpForwarding no
```

Sometimes you would enable SSH forwarding just for some users, for example the following lines enable it for foobar:

```
AllowTcpForwarding no  
Match User foobar  
AllowTcpForwarding yes
```

### Disable TCP forwarding

SSH supports “traffic tunneling”, it is used to forward TCP traffic over SSH channel. If you are not using this feature it is suggested to disable it. To disable TCP forwarding, edit *sshd\_config* file and disable *AllowTcpForwarding*:

```
AllowTcpForwarding no
```

### Disable user environment

Users logging via SSH are usually able to set environment options and potentially bypass some access restrictions. It is suggested, if this feature is not needed, to remove this permission, edit *sshd\_config* file and disable *PermitUserEnvironment*:



```
PermitUserEnvironment no
```

### Disable X11 forwarding

SSH supports X display forwarding, so X11 applications started on the remote system via SSH have their display shown on the client. If this feature is not used it is suggested to disable it, although it is disabled by default in most distributions. To disable X11 forwarding, edit `sshd_config` file and disable `X11Forwarding`:

```
X11Forwarding no
```

### Display a warning message before login

A pre login SSH banner shows before the password prompt, during an interactive session. It is usually used for legal warnings or to show the terms by which someone is allowed to use the system. This message is commonly located in `/etc/issue` but you can also use your custom file, for example `/etc/ssh/banner`. It is suggested to use a warning banner, edit `sshd_config` file and set `Banner` option:

```
Banner /etc/ssh/banner
```

### Do not use SSH Agent Forwarding

SSH Agent Forwarding is as an easy way to connect to a host with your SSH key and from there connect to another host with the same key. For example this is used when you cannot connect directly to the second host from your workstation. To enable SSH Agent Forwarding from command line you have to use `ssh -A` from command line or edit the `AgentForward` option in your SSH configuration file. It is suggested to not use SSH Agent Forwarding because it comes at cost of a security issue: a port-forwarding will be set up to connect you to the second host, so anyone with sufficient permission on the first host could be able to use that socket to connect to and use your local ssh-agent. It is recommended to never use SSH Agent Forwarding, if it is really needed by your use case it is suggested to use the option `ProxyCommand` instead.

### Hash Known Hosts

If a machine is compromised, a good idea is to minimize how much usable information is given to an attacker. The `known_hosts` file is a source of relevant information. `HashKnownHosts` is a configurable option, used to hash host names and addresses when they are added to `~/.ssh/known_hosts`. It is suggested to enable it, add it to your SSH configuration file:

```
HashKnownHosts Yes
```

### Key storage

It is suggested to store your SSH keys in a secure storage and always encrypt your key files using a strong password. For example, you may want to store them on a secure and encrypted pendrive and only plug it in when you want to use SSH.

### Increase Key Strength

It is suggested to use a length more than the default one. The following command instructs `ssh-keygen` with `-b` argument to generate a 4096-bit key:

```
$ ssh-keygen -b 4096 -t rsa -f ~/.ssh/id_rsa
```

Feel free to increase this to your desired key length although remember to use powers of two. To slow down cracking attempts it is suggested to iterate the hash function many times, for example iterating 6000 times using the `-a` option:

```
$ ssh-keygen -b 4096 -a 6000 -t rsa -f ~/.ssh/id_rsa
```

### Limit port forwarding

You don't want to expose the ports you open with port forwarding to other people. It is suggested to disable *GatewayPorts*, although in most distribution it is by default, to ensure that any port forwarding is limited to the local machine:

```
GatewayPorts no
```

### Limiting brute forcing attempts

SSH is a service target of worms, script kiddies, and all kind of brute forcing around. It's a good idea to limit the maximum amount of login tries for second. This can be achieved with a few iptables lines or with [DenyHosts](#).

### Message authentication codes

There are multiple ways to combine ciphers and MACs but only Encrypt-then-MAC should be used. It is suggested to use a selected list of MACs, edit `sshd_config` file:

```
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-ripemd160-  
↪ etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-ripemd160,  
↪ umac-128@openssh.com
```

Also set the same configuration for SSH client, edit `ssh_config` file:

```
Host *  
    MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-ripemd160-  
↪ etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-ripemd160,  
↪ umac-128@openssh.com
```

### OTP Setup

Usually SSH only verifies one thing, your password or your private key, although multiple authentication methods were allowed. Here we are going to see how to use Google Authentication as a OTP token during SSH authentication. Install the Google Authenticator PAM module, for example in Ubuntu you can use this command:

```
apt-get install libpam-google-authenticator
```

Run the command `google-authenticator` for each user you need an OTP token on your device, you will get some questions to configure the token generator and at the end, a QR code will be displayed. Use it to setup your access token, for example on your phone, and safely save all the codes displayed.

Configure SSH to use PAM editing *sshd\_config* file with these values:

```
ChallengeResponseAuthentication yes
PasswordAuthentication no
AuthenticationMethods publickey,keyboard-interactive
UsePAM yes
PubkeyAuthentication yes
```

Restart the SSH service. Now edit the PAM configuration to use Google Authentication, edit */etc/pam.d/sshd* and replace the line:

```
@include common-auth
```

With the line:

```
auth required pam_google_authenticator.so
```

Now SSH logins will require a private key, and after it will additionally require an OTP token. Log in as the user you'll be logging in with remotely and run the *google-authenticator* command to create a secret key for that user. Restart SSH daemon.

## Restrict IP Listen Address

If you are in a multi homed setup (with multiple network interfaces) it is suggested to avoid having SSH listening on all interfaces, unless it is really needed. For example only a specific IP should be used for SSH. To specify on which IP to listen, edit *sshd\_config* file use *ListenAddress* option, for example to listen only on the interface with IP 192.168.0.1:

```
ListenAddress 192.168.0.1
```

## Reduce Grace Time

It is suggested to lower the default grace time for authenticating a user, it is only necessary if you are on a very slow connection otherwise it will hold unauthenticated connections open for some time. To reduce the *gracetime* to 30 seconds, edit *sshd\_config* file use *LoginGraceTime* option:

```
LoginGraceTime 30
```

## Route traffic over TOR

If you would like to provide an additional layer of encryption, server authentication and some traffic analysis resistance you can access your SSH as an hidden service over TOR. Note: Attackers can still attack the SSH service, but don't know who they are attacking. This hardening step is not suggested, only a desiderata in needs of mention.

If you want to access your SSH daemon only via hidden service, bind it only to localhost, edit *sshd\_config*:

```
ListenAddress 127.0.0.1:22
```

Create youe hidden service editing *torrc* (usually in */etc/tor/torrc*):

```
HiddenServiceDir /var/lib/tor/hidden_service/ssh
HiddenServicePort 22 127.0.0.1:22
```

You will find the hostname you have to use in `/var/lib/tor/hidden_service/ssh/hostname`. Now you have to configure SSH client to connect over TOR. Install *socat* (it is used to route traffic over TOR) and configure SSH to use *socat* for each domain ending with *.onion*, editing *ssh\_config*:

```
Host *.onion
    ProxyCommand socat - SOCKS4A:localhost:%h:%p,socksport=9050
```

### Symmetric ciphers

Symmetric ciphers are used to encrypt the transmission after the initial key exchange and successful authentication.

It is suggested to use a selected list of strong ciphers, edit *sshd\_config* file:

```
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,
↪aes256-ctr,aes192-ctr,aes128-ctr
```

Also set the same configuration for SSH client, edit *ssh\_config* file:

```
Host *
    Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.
↪com,aes256-ctr,aes192-ctr,aes128-ctr
```

### Use PAM

By default, OpenSSH uses PAM for the authentication of users. PAM (Pluggable Authentication Modules) is a powerful framework for managing authentication of users. Using PAM you can enforce rules during the authentication (i.e. limiting access based on login count). It is suggested to use PAM for SSH authentication too, edit *sshd\_config* file and enable *UsePAM*:

```
UsePAM yes
```

### Use privilege separation

It is a good practice to never run processes as root, if you enable SSH privilege separation, the SSHd process has a tiny footprint running as root and it drops privileges as soon as possible to run as unprivileged process. It is suggested to enable privilege separation (usually it is enabled by default), edit *sshd\_config* file and enable *UsePrivilegeSeparation*:

```
UsePrivilegeSeparation yes
```

### Use strong key algorithms

SSH supports different key exchange algorithms, ciphers and message authentication codes. There are ciphers for any security level. It is suggested to use only strong key exchange protocols, edit *sshd\_config* file and set *KexAlgorithms*:

```
KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
```

Edit *ssh\_config* file and set *KexAlgorithms*:

```
# Github needs diffie-hellman-group-exchange-sha1 some of the time but not always.
#Host github.com
#    KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256,
↪diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1
```

(continues on next page)

(continued from previous page)

```
Host *
    KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
```

Open `/etc/ssh/moduli` if exists, and delete lines where the 5th column is less than 2000:

```
awk '$5 > 2000' /etc/ssh/moduli > "${HOME}/moduli"
wc -l "${HOME}/moduli" # make sure there is something left
mv "${HOME}/moduli" /etc/ssh/moduli
If it does not exist, create it:

ssh-keygen -G "${HOME}/moduli" -b 4096
ssh-keygen -T /etc/ssh/moduli -f "${HOME}/moduli"
rm "${HOME}/moduli"
```

## Whitelisting / blacklisting users

By default all systems user can login via SSH using their password or public key. Sometime you create UNIX / Linux user account for ftp or email purpose. However, those user can login to system using SSH. To only allow antani and tapioco user to use the system via SSH, add the following to `sshd_config`:

```
AllowUsers antani tapioco
```

Alternatively, you can allow all users to login via SSH but deny only a few users, with the following line:

```
DenyUsers foo bar
```

You can also configure Linux PAM allows or deny login via the sshd server.

## Whitelisting / blacklisting groups

By default all systems user can login via SSH using their password or public key. Sometime you create UNIX / Linux user account for ftp or email purpose. However, those user can login to system using SSH. To only allow users in a group (fo example in the foo group), add the following to `sshd_config`:

```
AllowGroups foo
```

Alternatively, you can allow all users to login via SSH but deny only the users in the foo group, with the following line:

```
DenyGroups foo
```

You can also configure Linux PAM allows or deny login via the sshd server.

## References

- <https://heipei.github.io/2015/02/26/SSH-Agent-Forwarding-considered-harmful/>
- <https://stribika.github.io/2015/01/04/secure-secure-shell.html>

### 1.1.11 OpenVPN

According to [OpenVPN official website](#) OpenVPN is “an open-source software that implements virtual private network (VPN) techniques for creating secure point-to-point connections”.

- *Additional authentication*
- *Attacks on default gateway*
- *Custom Port*
- *Disable IPv6*
- *Disable management interface*
- *DNS management*
- *Don not allow certificate re-use*
- *Key Size*
- *Limit concurrent clients*
- *Persistent VPN device*
- *Run as unprivileged user*
- *Secure Ciphers*
- *Secure PKI Management*
- *Set minimum TLS version*
- *SHA-2 for message authentication*
- *Use PSK*
- *Verify Certificate subject name*
- *Verify CRL*
- *Verify the server certificate*
- *References*

#### Additional authentication

If possible, it is suggested to request an additional authentication in addition to a client certificate. This could protect you in case of certificate loss. Additional authentication could be configured server side in two ways:

- Using the *auth-user-pass-verify*
- Using a plugin (i.e. PAM)

#### Attacks on default gateway

OpenVPN is commonly used to route all traffic or only some subnets through the VPN tunnel. This is implemented adding wide scope routing rules. A rogue DHCP server able to push more specific routes could be able to take precedence on the routing table and route your traffic outside the VPN. To prevent this kind of attacks it is suggested to configure your DHCP client to ignore classless static routes. A rogue DHCP could also push a subnet mask for an extremely large subnet, so all the traffic could be routed on the local network and not in the VPN. This issue has not an

easy solution, it depends by your OS, for example in Linux you can use advanced routing and multiple routing table (see [https://www.agwa.name/blog/post/hardening\\_openvpn\\_for\\_def\\_con](https://www.agwa.name/blog/post/hardening_openvpn_for_def_con)).

### Custom Port

It is suggested to move OpenVPN from the default port to a custom one. For example we are setting it on port 10000, edit the server configuration file as follows:

```
port 10000
```

### Disable IPv6

You know, IPv6 could be a security beast. Unless you are using IPv6 in your OpenVPN tunnel, then all IPv6 traffic from your client will bypass the VPN and egress over the local network. It is suggested to disable IPv6 support in your OS if you are not using it.

### Disable management interface

The OpenVPN Management interface allows OpenVPN to be remotely administered. It is suggested to disable or restrict to localhost (or local trusted clients) the management interface. Edit the server configuration file and comment the *management* option or make sure it is only accessible via localhost:

```
# management 127.0.0.1 8000
```

### DNS management

When you are using a VPN tunnel, you should use only a trusted DNS server. If an attacker is able to push a rogue DNS server it is a game over for you because he could redirect all your traffic outside the VPN. It should take care of your configured DNS servers, unfortunately how DHCP clients manage pushed DNS servers depends by operating systems. Some systems do it incredibly poorly and it is possible to change your DNS server, by pushing it via DHCP, after the VPN tunnel startup. It is suggested to pin your DNS servers to be sure you are always using the right one.

### Don not allow certificate re-use

Certificates should not be shared and each VPN client must have its unique certificate. It is suggested to enforce it disabling the *duplicate-cn* in the server configuration file, if present, commenting or deleting it, as follows:

```
# duplicate-cn
```

### Key Size

It is suggested to use a key size of, at least 2048 bits (better 4096 bits), for your certificates. If you are creating certificates with OpenSSL you have to modify the *default\_bits* parameter.

### Limit concurrent clients

It is suggested to restrict the maximum number of concurrent clients to a reasonable number. Set *max-clients* in the server configuration file, as follows (limited at 100 clients in the example):

```
max-clients 100
```

### Persistent VPN device

If your connection is interrupted and OpenVPN is trying to reconnect, in the meanwhile, traffic is passing by your default route, bypassing your VPN. It is suggested to configure OpenVPN to keep the device open and to hold traffic until the connection is restored, add the following option to the configuration file:

```
persist-tun
```

### Run as unprivileged user

It is suggested to run OpenVPN process as unprivileged user. Add the following lines to the configuration file:

```
user nobody  
group nobody
```

### Secure Ciphers

It is suggested to use strong symmetric ciphers (at least 256bit). For example, add to both server and client configuration file the following to use AES-256:

```
cipher AES-256-CBC
```

Is also suggested to limit the use of TLS ciphersuites with:

```
tls-cipher TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256:TLS-ECDHE-ECDSA-WITH-AES-128-GCM-  
↪SHA256:TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-SHA256
```

### Secure PKI Management

OpenVPN authentication, in most cases, is based on PKI and X.509 certificates. Practicing secure PKI management is mandatory to safeguard, also, OpenVPN. It is suggested to follow best practices for secure PKI management, for example:

- Secure management of CA PKI.
- Generate private keys on the target system and never transport them.
- Never share private keys.
- Use certificate passwords if possible and use a secure password policy.
- Use a CRL and revoke lost/compromised keys.



## Set minimum TLS version

It is suggested to set minimum TLS version editing the configuration file and adding:

```
tls-version-min 1.2
```

## SHA-2 for message authentication

It is suggested to use strong algorithm for message authentication (HMAC). Add the following line to the configuration file:

```
auth SHA-256
```

## Use PSK

The `-tls-auth` option uses a static pre-shared key (PSK) shared among all connected peers. This is an extra layer of protection to the TLS channel by requiring that incoming connections are correctly HMAC signed by the PSK key. This feature could protect your VPN server by DoS attacks aimed to load your CPU load, by port scanning avoiding service fingerprinting, and act as second line of defense for SSL library vulnerabilities. Generate a PSK with the command:

```
openvpn --genkey --secret ta.key
```

Add the following line to your server configuration:

```
tls-auth ta.key
```

Add the following line to your server configuration:

```
tls-auth ta.key
```

Beware, the `-tls-auth` key is changed, it must be changed on all peers at the same time, so it could potentially lead to a network management horror story. It is suggested to use it with care.

## Verify Certificate subject name

This is not a general recommendation although in some cases could be useful to verify X.509 certificate subject name on the client. Add to the client configuration file the following line:

```
verify-x509-name 'C=XX, O=Example, CN=example.xxx' subject
```

## Verify CRL

It is suggested to verify revoked client certificates, they should not connect or keep a connection alive. Add `crl-verify` to the server configuration file, as follows:

```
crl-verify path/yourcrl.pem
```

### Verify the server certificate

It is recommended to check that the server certificate contains a specific key usage and an extended key usage. Add to the client configuration file the following line:

```
remote-cert-tls server
```

This also is a measure to prevent a client using his certificate to impersonate a server.

Certificates using the X509v3 format have key usage flags set. Clients should use certificates with the “TLS Web Client Authentication” set and servers with “TLS Web Server Authentication” set.

Add to the client configuration file the following line:

```
remote-cert-eku "TLS Web Server Authentication"
```

Add to the server configuration file the following line:

```
remote-cert-eku "TLS Web Client Authentication"
```

### References

- <https://community.openvpn.net/openvpn/wiki/Hardening>
- <http://darizotas.blogspot.it/2014/04/openvpn-hardening-cheat-sheet.html>
- [https://www.agwa.name/blog/post/hardening\\_openvpn\\_for\\_def\\_con](https://www.agwa.name/blog/post/hardening_openvpn_for_def_con)

## 1.2 Operating System Hardening

This chapter is about OS hardening.

### 1.2.1 MacOS 10.12 Sierra

According to [Wikipedia](#) Sierra is “(version 10.12) is the thirteenth major release of macOS (previously OS X), Apple Inc.’s desktop and server operating system for Macintosh computers. The successor to OS X El Capitan, it is the first version of the operating system issued under the June 2016 rebranding as macOS. Sierra is named after California’s Sierra Nevada mountain range. Its major new features concern Continuity, iCloud, and windowing, as well as support for Apple Pay and Siri.”.

- *Applications*
- *Allow only signed apps*
- *Check Privacy permissions*
- *Destroy FileVault Keys*
- *Disable Creation of Metadata Files*
- *Disable Diagnostics*
- *Disable Guest user*
- *Disable Handoff*

- *Disable password hints*
- *Disable recent items*
- *Disable Localization Services*
- *Disable Spotlight Suggestions*
- *Enable FileVault*
- *Enable Firewall*
- *Enable screen saver*
- *Empty trash securely*
- *Erase free space*
- *Homebrew hardening*
- *Power off memory during standby*
- *Require an administration password*
- *Require password to un-lock*
- *Save to Disk by Default*
- *Set a Firmware Password*
- *Show all filename extensions*
- *Show when localization is used*
- *Users privilege separation*
- *References*

## Applications

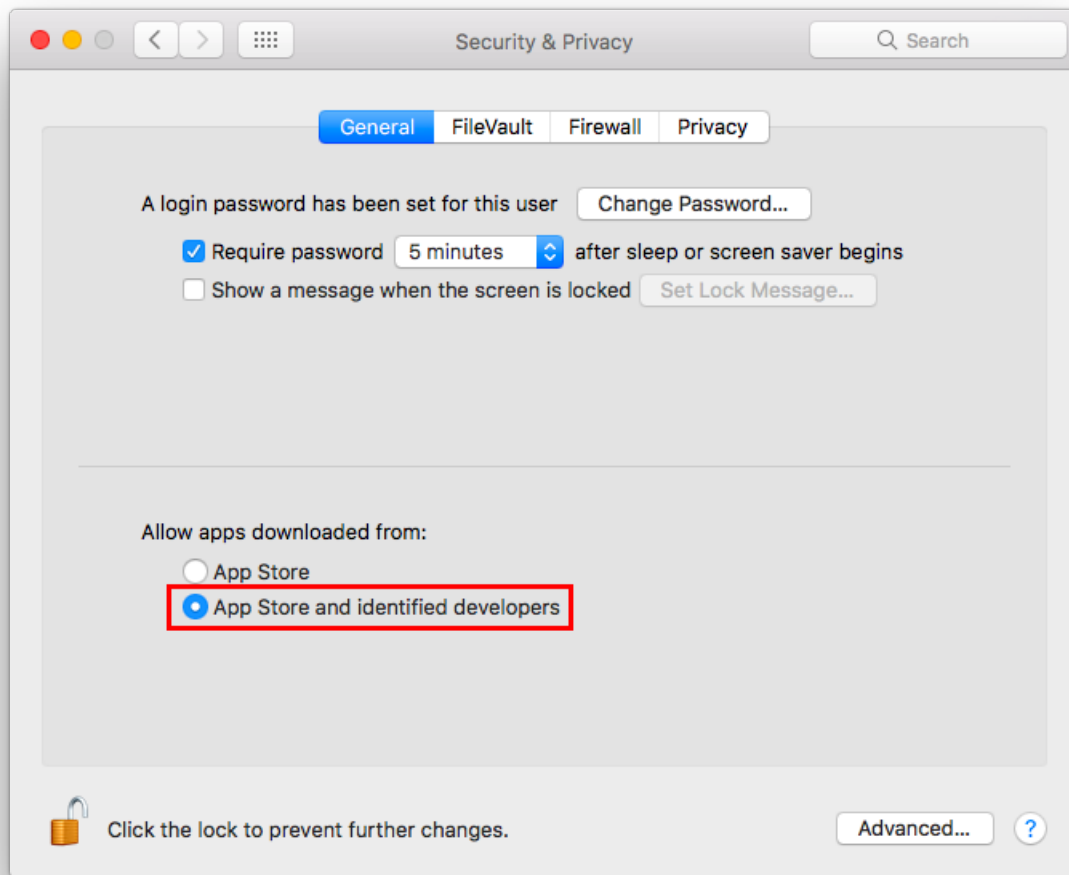
It is suggested to keep the */Applications/* directory as clean as possible, having a separate *Applications* directory for your apps. Just create a folder named “Applications” in your home directory (or where you like) and install (move) all applications there. Apps installed via App Store or some special apps cannot live in a custom Applications folder, so you have to keep them in the original Applications.

### Allow only signed apps

It is suggested to never run untrusted code not signed with a proper key. To allow only apps signed by an authorized developer, go to:

System Preferences Security & Privacy General

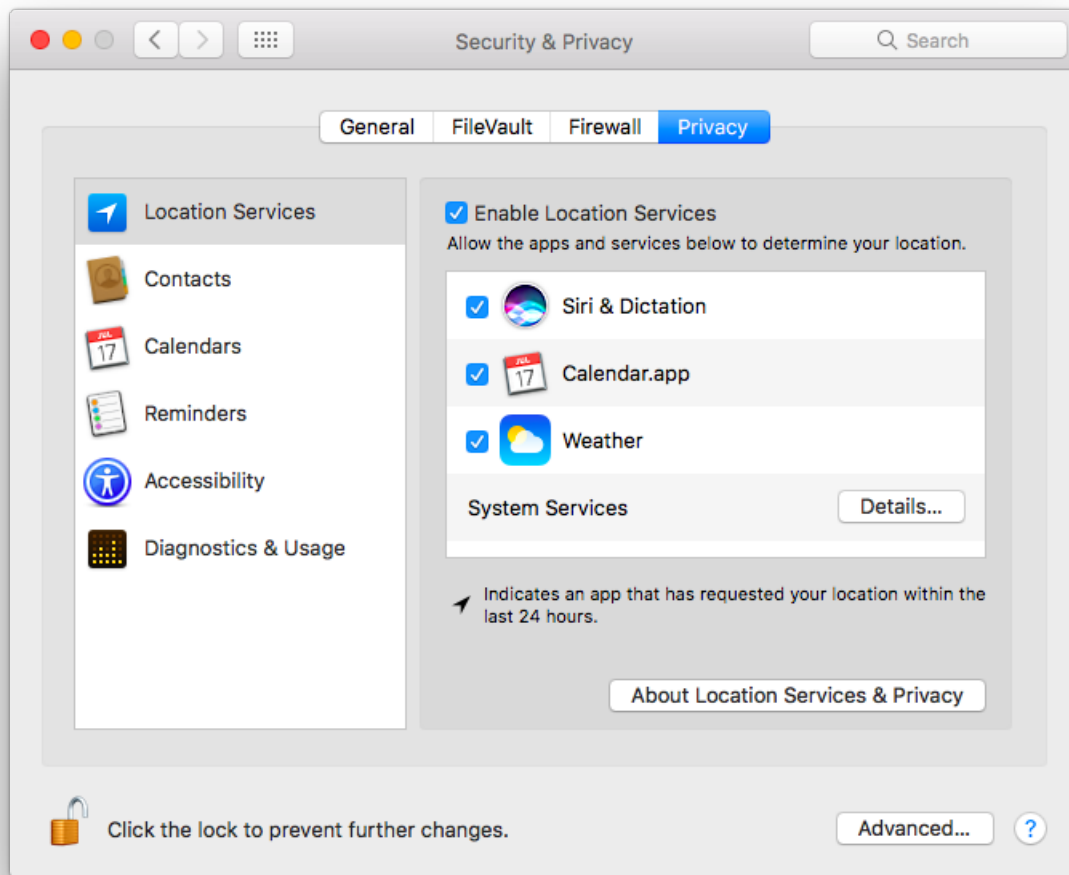
Set “Allow apps download from” to “Mac App Store and identified developers” or if you want to be more strict and you install applications only via App Store set it to “Mac App Store”. In OS X Sierra is now not possible to choose to run unsigned code, it was in OS X El Capitan.



### Check Privacy permissions

OS X allows you to track all applications requesting access to some sort of sensitive data, for example your location or your contacts. It is suggested to periodically check the list of applications requesting access to sensitive data and review their permissions. To show the list of these applications go to:

System Preferences Security & Privacy Privacy



## Destroy FileVault Keys

By default File Vault keys are kept when system goes in standby mode. As suggested by *man pmset*:

**destroyfvkeyonstandby - Destroy File Vault Key when going to standby** mode. By default File vault keys are retained even when system goes to standby. If the keys are destroyed, user will be prompted to enter the password while coming out of standby mode.(value: 1 - Destroy, 0 - Retain)

It is suggested to configure your system to destroy File Vault keys when entering in standby mode with the following command:

```
sudo pmset destroyfvkeyonstandby 1
```

## Disable Creation of Metadata Files

By default OS X creates metadata files in each directory to speed up browsing. These files could leak metadata, it is suggested to avoid creation of .DS\_Store and AppleDouble files.

Disable Creation of Metadata Files on Network Volumes with the following command in a Terminal:

```
defaults write com.apple.desktopservices DSDontWriteNetworkStores -bool true
```

Disable Creation of Metadata Files on USB Volumes with the following command in a Terminal:

```
defaults write com.apple.desktopservices DSDontWriteUSBStores -bool true
```

### Disable Diagnostics

It is suggested to disable diagnostic data and usage data to Apple. Go to:

System Preferences Security & Privacy Privacy Diagnostics & Usage

Un-check “Send diagnostic & usage data to Apple”. Un-check “Share crash data with app developers”.

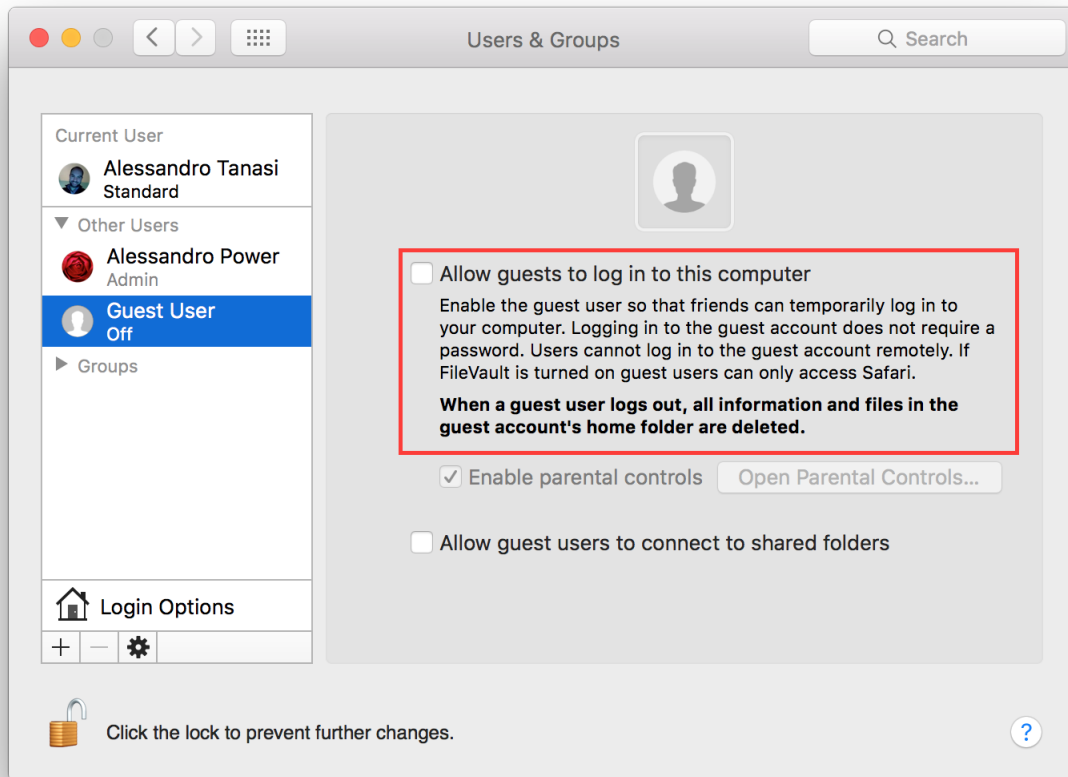


### Disable Guest user

Mac OS X comes with a *Guest* user enabled by default, it permits the use of your device in a restricted environment to anyone. It is suggested to disable the *Guest* user, go to:

System Preferences Users & Groups Guest User

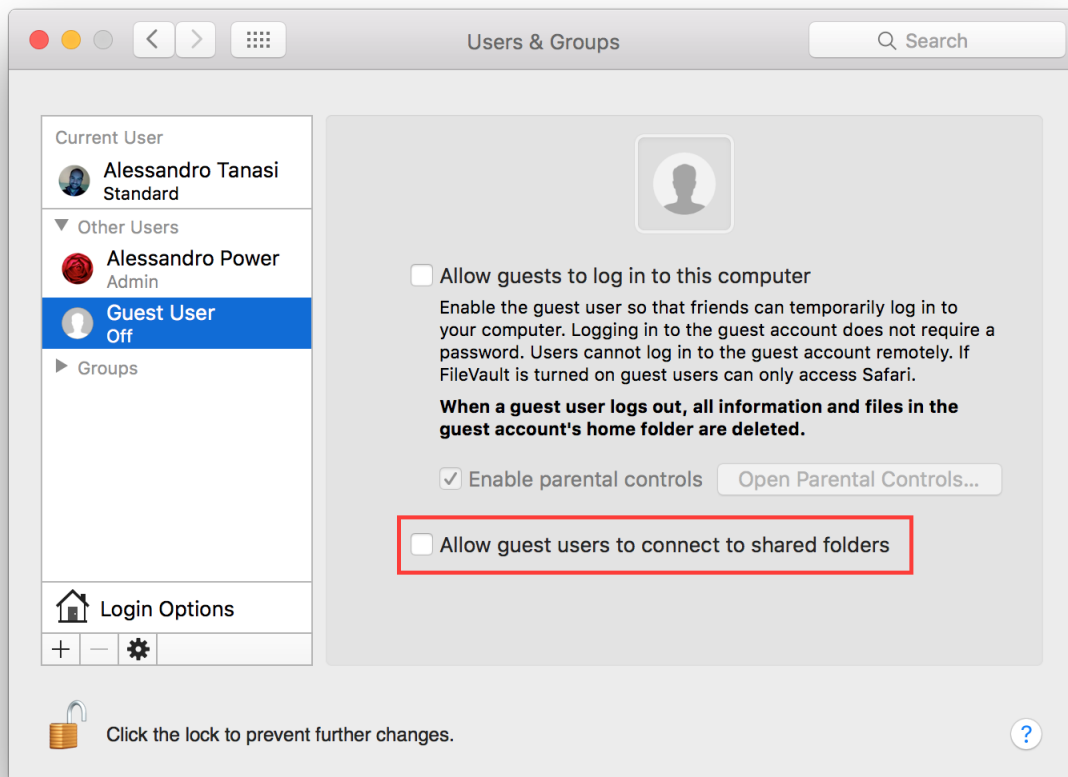
Un-check “Allow guests to log in to this computer”.



It is suggested to disable guest access to shared folders, if you are not using it, go to:

System Preferences Users & Groups Guest User

Un-check “Allow guest users to connect to shared folders”.



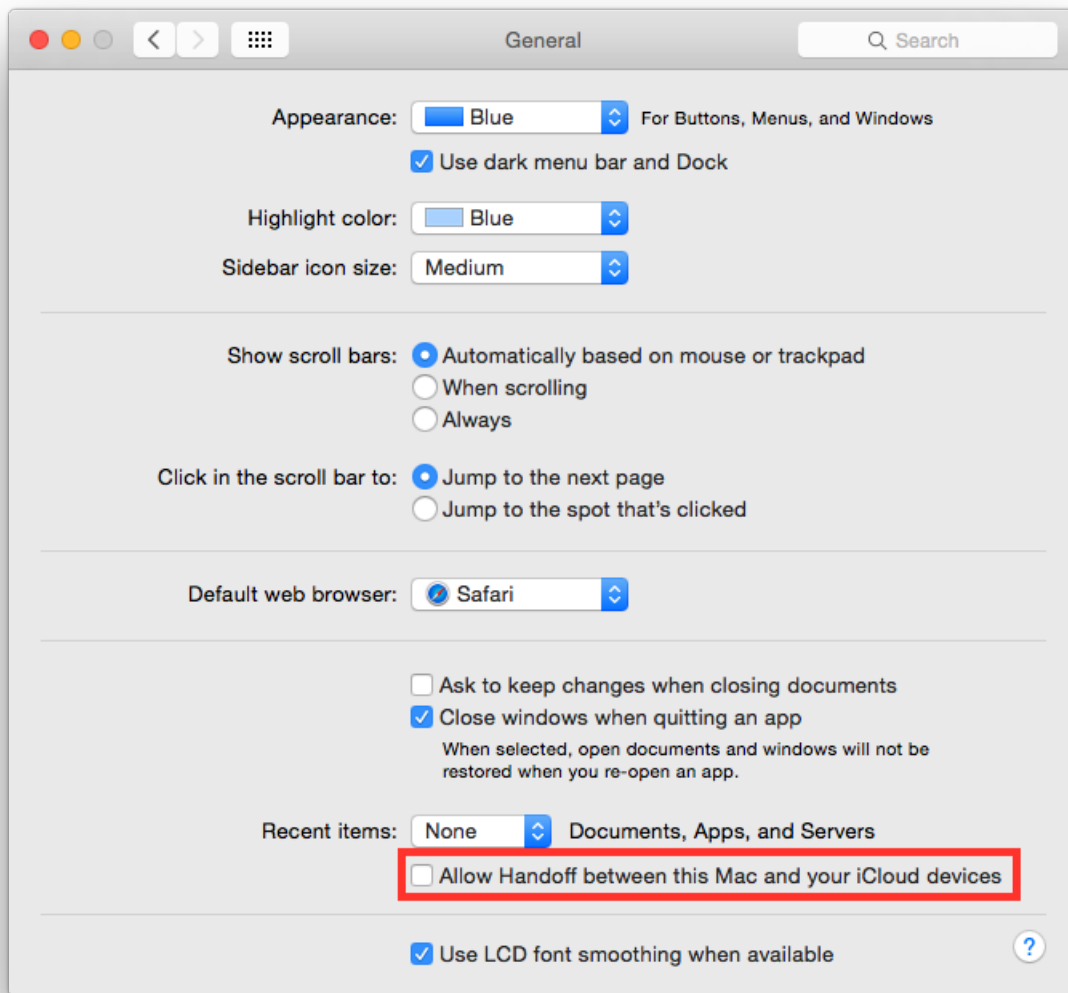
### Disable Handoff

Handoff is a great feature to keep your work in sync between Apple devices. Due to its implementation it needs to send some data to Apple iCloud to work, so in some way it is leaking your data. It is suggested to disable it. Go to:

System Preferences General

Un-check “Allow Handoff between this Mac and your iCloud devices”.



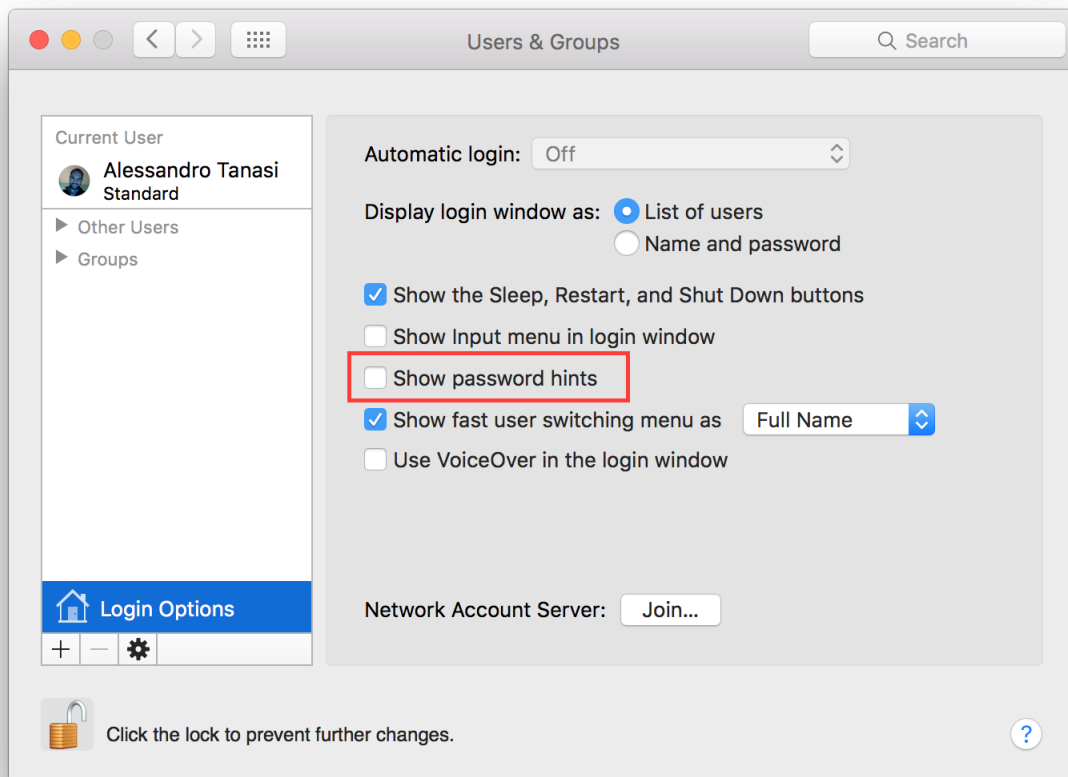


## Disable password hints

Passwords hints are supposed to help an user to remember his password but could also help attackers. It is suggested to disable password hints, go to:

System Preferences Users & Groups Login Options

Un-check "Show password hints".

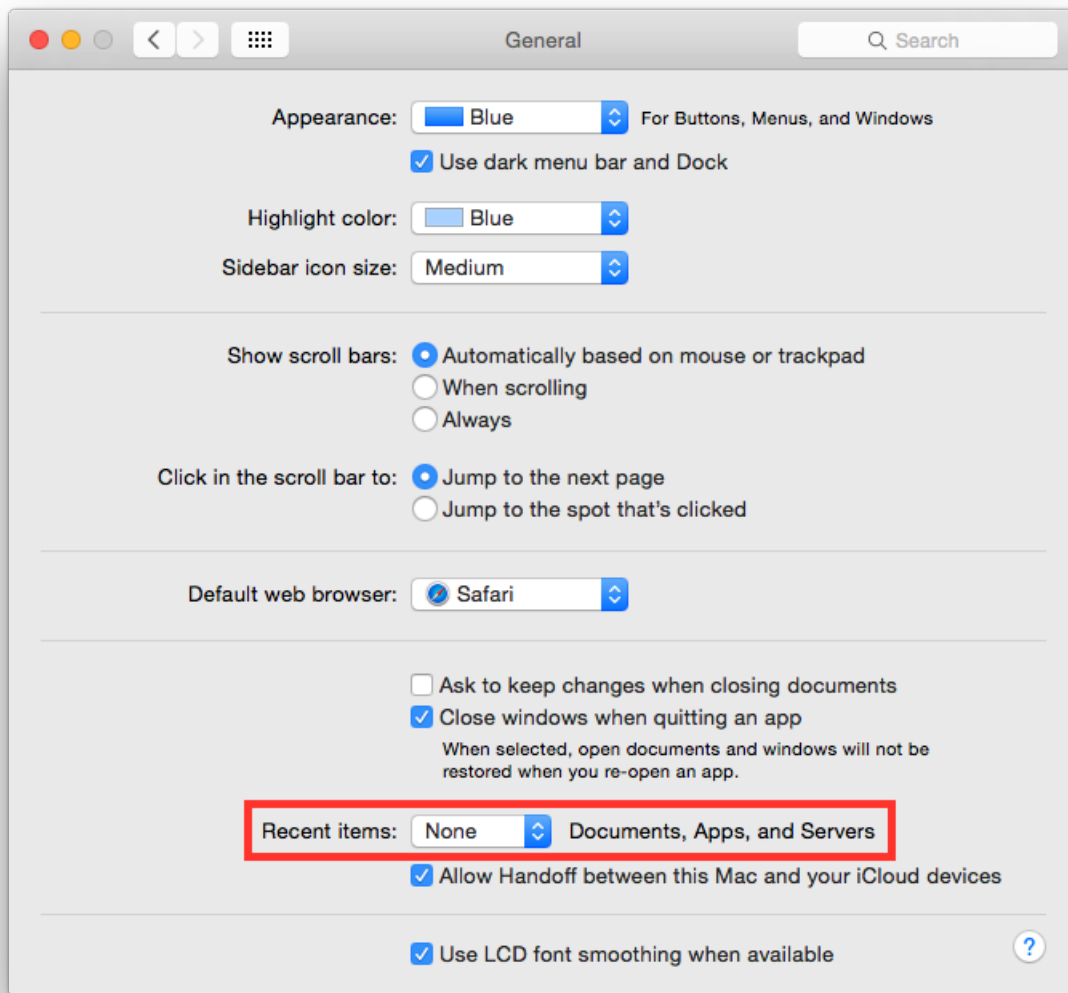


### Disable recent items

Recent items are used to track your latest activity, it is also a feature used in forensics investigation to create the user activity timeline. It is suggested to not track last recently used items. Go to:

System Preferences General

Set “Recent items” to “None”.

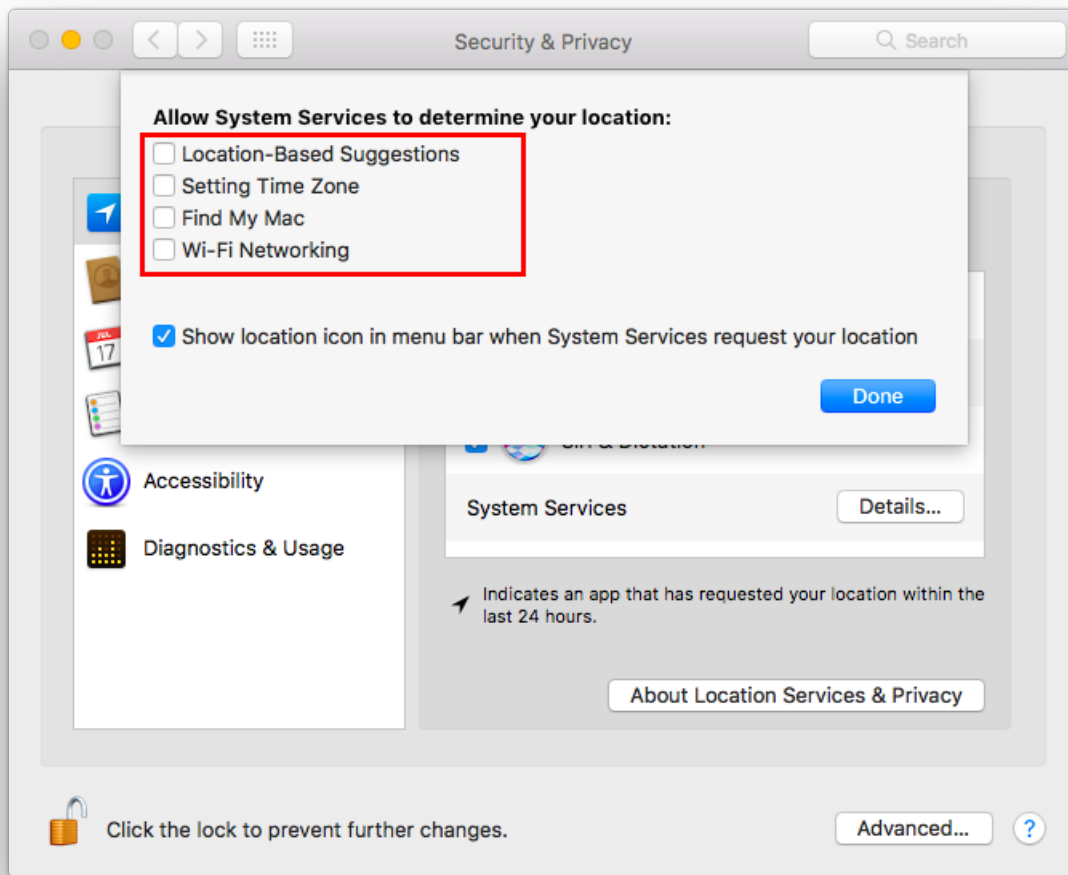


## Disable Localization Services

By default Spotlight is allowed to use localization services to help you offering localized results. Due to his implementation it needs to send your position to a remote service. It is suggested to disable this behavior. Go to:

System Preferences Security & Privacy Privacy Location Services

Select “System Services” and click “Details...”. It is suggested to disable localization for all services, if not needed.

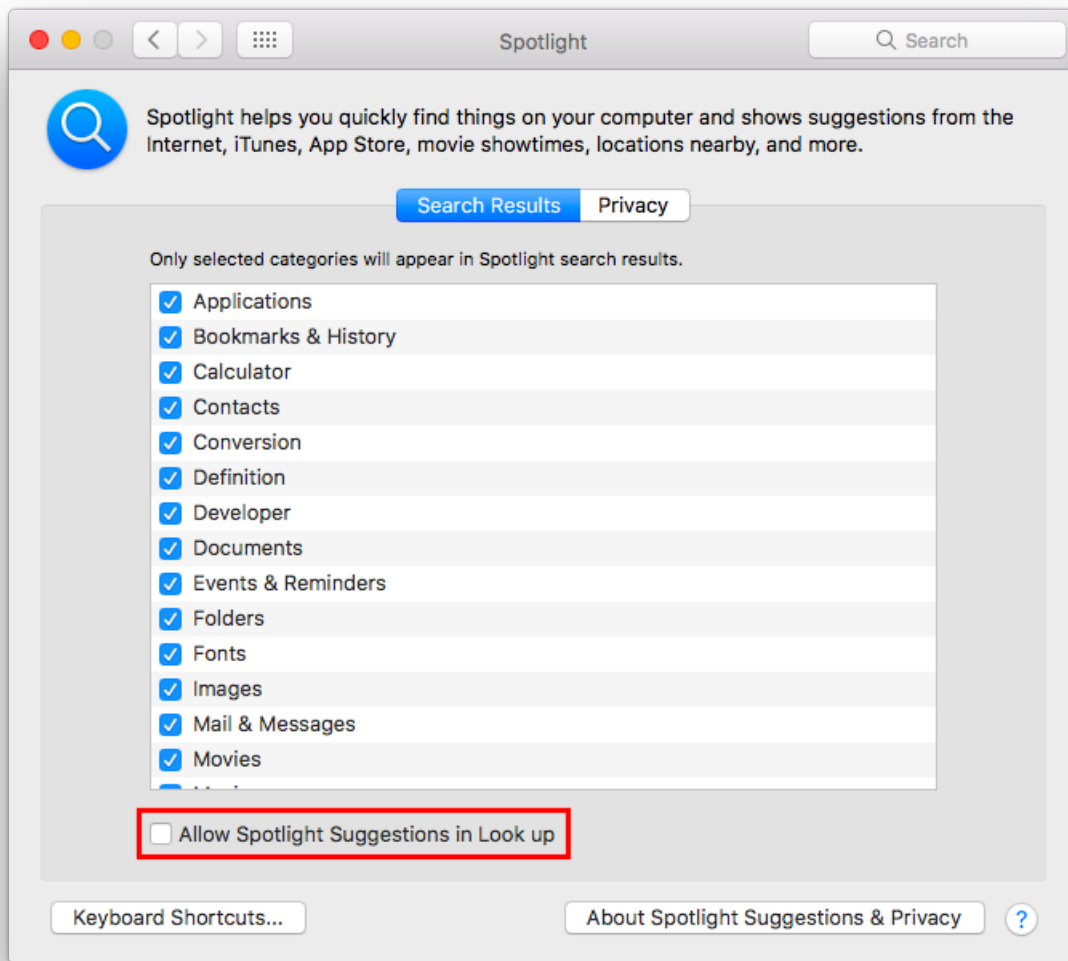


## Disable Spotlight Suggestions

By default Spotlight shows suggestions from the Internet, it sends your search to Apple services and provides results back. It is suggested to use Spotlight only locally to prevent leaking your search. To disable Spotlight Suggestions go to:

System Preferences Spotlight

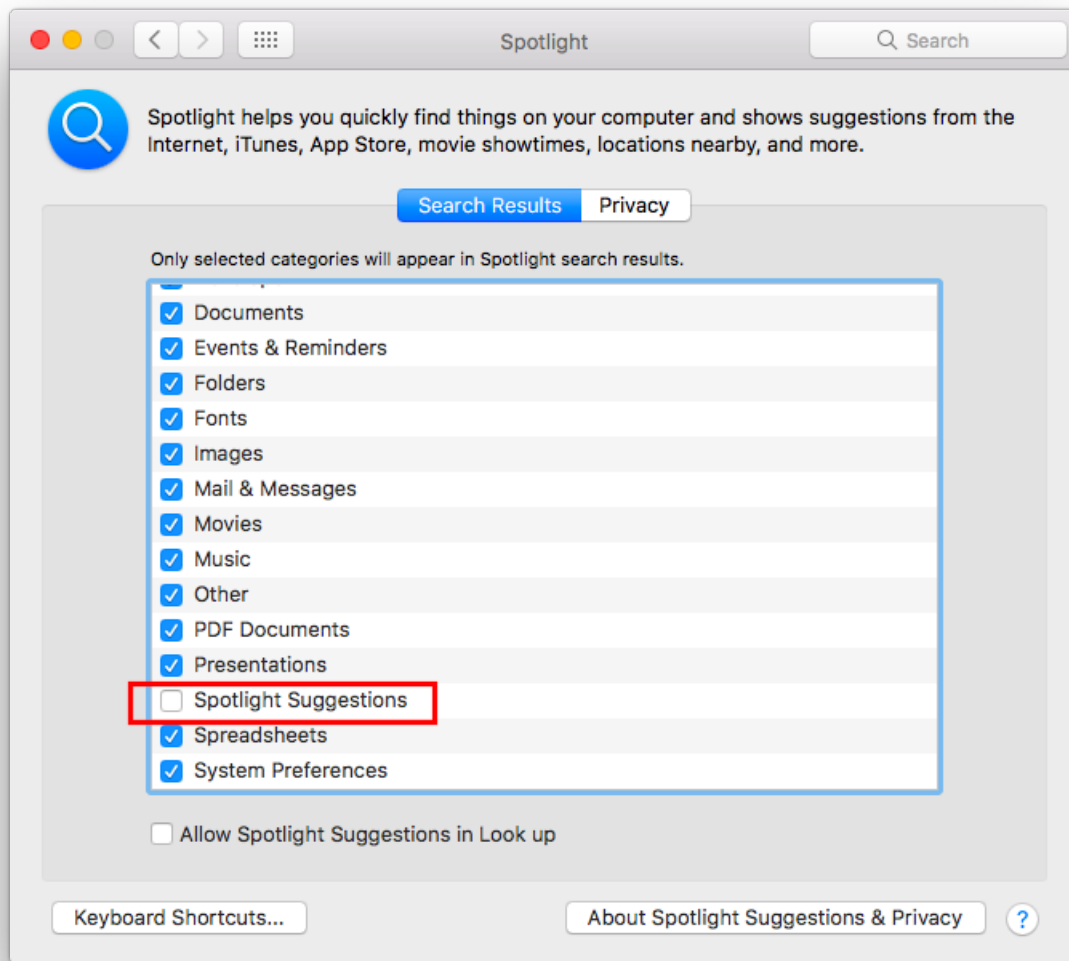
Un-check "Allow Spotlight Suggestions in Spotlight and Look Up".



It is suggested to disable Spotlight Suggestions to avoid leaking your search to online services used for suggestions, go to:

System Preferences Spotlight

Un-check "Spotlight Suggestions" from the list of results categories.



### Enable FileVault

It is suggested to enable FileVault to enable full disk encryption on your device. It should be already enabled by default. Go to:

System Preferences Security & Privacy FileVault

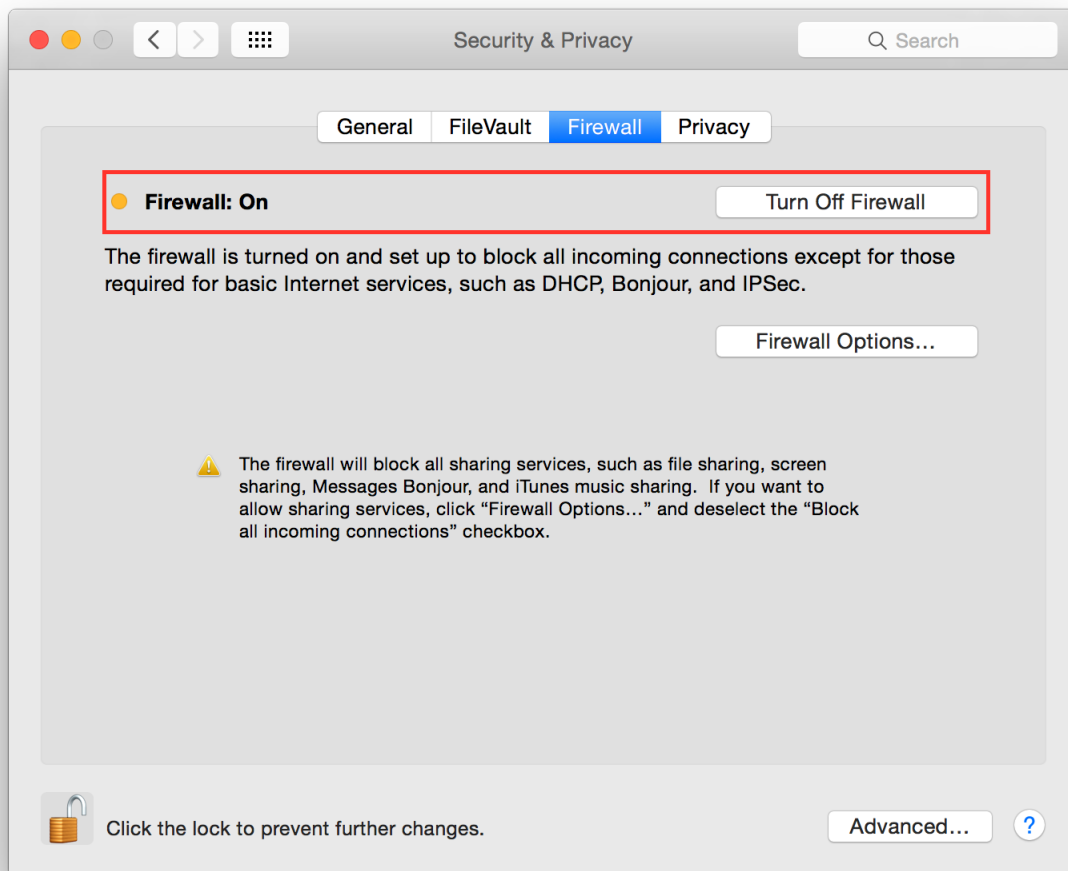
Enable FileVault.

### Enable Firewall

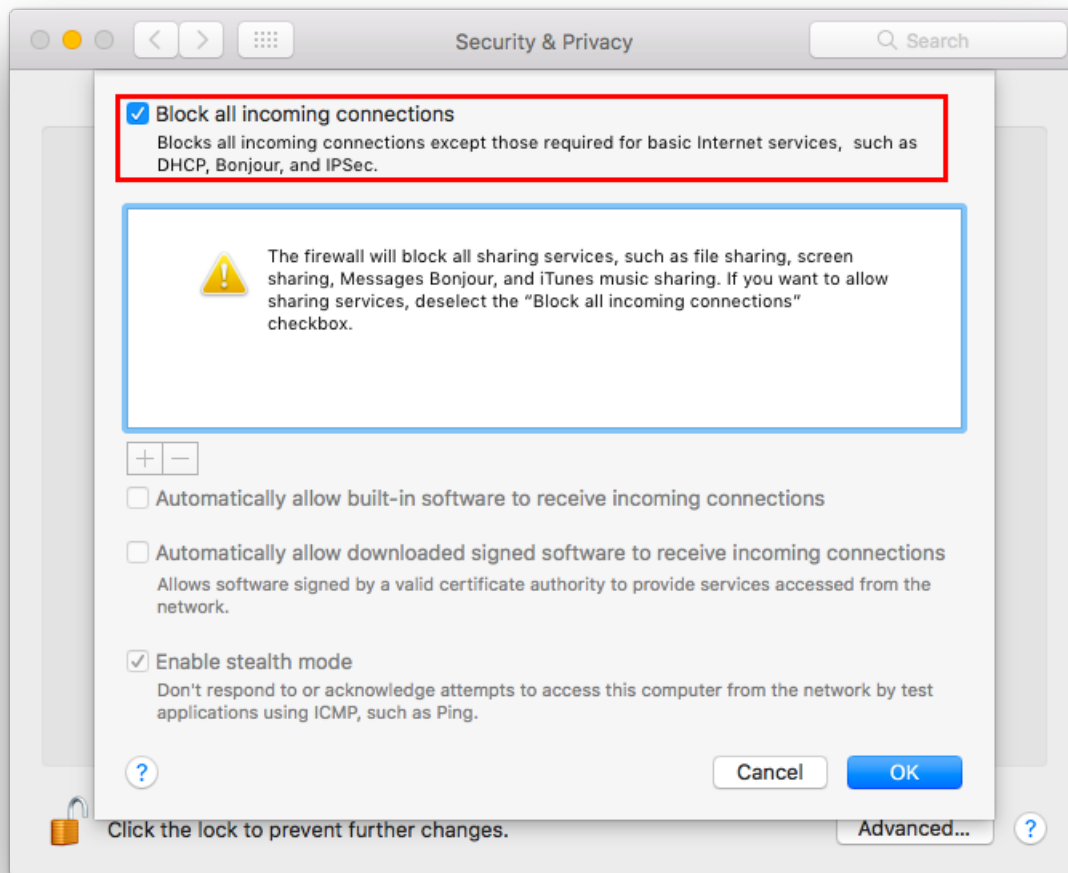
It is suggested to enable the Firewall and have it always running. Go to:

System Preferences Security & Privacy Firewall

Click on "Turn On Firewall".



Now click on "Firewall options", a new panel will appear. Click on "Block all incoming connections".



Using “Block all incoming connections” will block all incoming connections to your host. This will block also all sharing services, such as file sharing, screen sharing, Messages Bonjour, iTunes music sharing and other features. If your host is providing any kind of service, this option is not suggested; you should disable it.

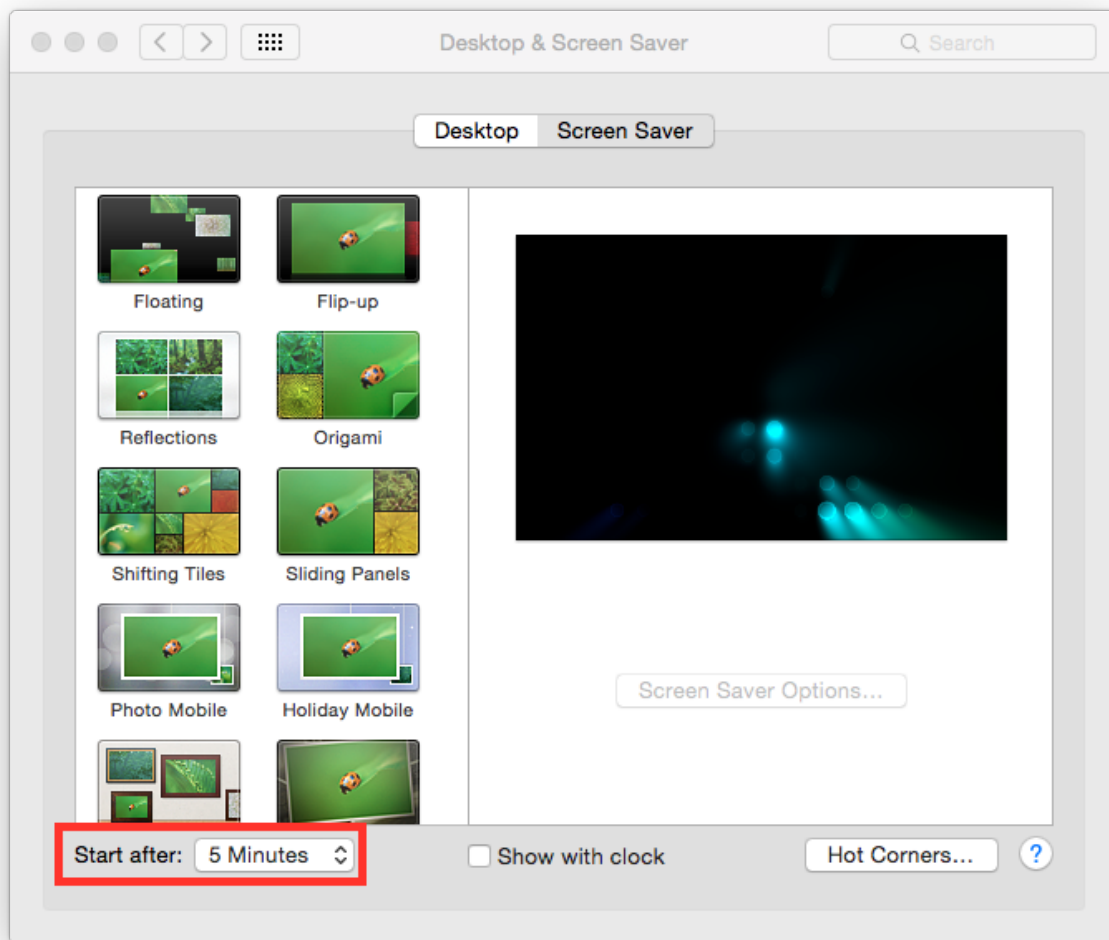
### Enable screen saver

It is suggested to enable the screen saver to automatically lock your screen after a while. Go to:

System Preferences Desktop & Screen Saver Screen Saver

Set “Start after” to “5 Minutes”.





## Empty trash securely

When you delete a file, OS X only deletes the index entry for the file, which tells the system the file's contents are free to be overwritten; however, the data still remains and may be recovered using a forensics software. It is a good practice to always empty your trash securely. Your data will be securely wiped from disk in an irreversible way. In the previous OS X releases there was an option to enable safe delete, Apple has removed this feature in OS X El Capitan. However, you can use command line tools.

You can use the `rm` command from Terminal to delete files with the `-P` option, as stated in `man rm` this option is used to:

Overwrite regular files before deleting them. Files are overwritten three times, first with the byte pattern 0xff, then 0x00, and then 0xff again, before they are deleted.

For example if you want to delete `test.pdf` you should open Terminal and use:

```
$ rm -P test.pdf
```

### Erase free space

In some cases, you might want to run an overwrite task on the free space of a given drive. You can use the *diskutil* command line utility, open Terminal and use:

```
diskutil secureErase freespace LEVEL /Volumes/DRIVE_NAME
```

In this command, change LEVEL to a number of 0 through 4, the available options are:

- 0 is a single-pass of zeros
- 1 is a single-pass of random numbers
- 2 is a 7-pass erase
- 3 is a 35-pass erase
- 4 is a 3-pass erase

Change DRIVE\_NAME to the name of the mount point.

### Homebrew hardening

Homebrew is a quite common third party tool in OS X systems.

It is suggested to disable anonymous statics collections adding the following variable to your *.bash\_profile* or *.profile* (or your shell configuration) file:

```
export HOMEBREW_NO_ANALYTICS=1
```

It is suggested to disable automatic updates to keep in control of brew updates, add the following to your *.bash\_profile* or *.profile* (or your shell configuration) file:

```
export HOMEBREW_NO_AUTO_UPDATE=1
```

It is suggested to configure brew to do not leak your GitHub username. When checking out a public repository, by default, your username is always sent. Add the following to your *.bash\_profile* or *.profile* (or your shell configuration) file:

```
export HOMEBREW_NO_GITHUB_API=1
```

It is suggested to configure brew to avoid protocol downgrades from HTTPS to HTTP via redirect. Add the following to your *.bash\_profile* or *.profile* (or your shell configuration) file:

```
export HOMEBREW_NO_INSECURE_REDIRECT=1
```

### Power off memory during standby

By default during stand-by memory are kept powered on, this is prone to forensics acquisition of your memory. As stated in *man pmset*:

hibernatemode supports values of 0, 3, or 25. Whether or not a hibernation image gets written is also dependent on the values of standby and autopoweroff

For example, on desktops that support standby a hibernation image will be written after the specified standbydelay time. To disable hibernation images completely, ensure hibernatemode standby and autopoweroff are all set to 0.

hibernatemode = 0 by default on desktops. The system will not back memory up to persistent storage. The system must wake from the contents of memory; the system will lose context on power loss. This is, historically, plain old sleep.

hibernatemode = 3 by default on portables. The system will store a copy of memory to persistent storage (the disk), and will power memory during sleep. The system will wake from memory, unless a power loss forces it to restore from hibernate image.

hibernatemode = 25 is only settable via pmset. The system will store a copy of memory to persistent storage (the disk), and will remove power to memory. The system will restore from disk image. If you want “hibernation” - slower sleeps, slower wakes, and better battery life, you should use this setting.

It is suggested to power off memory at stand-by with the following command:

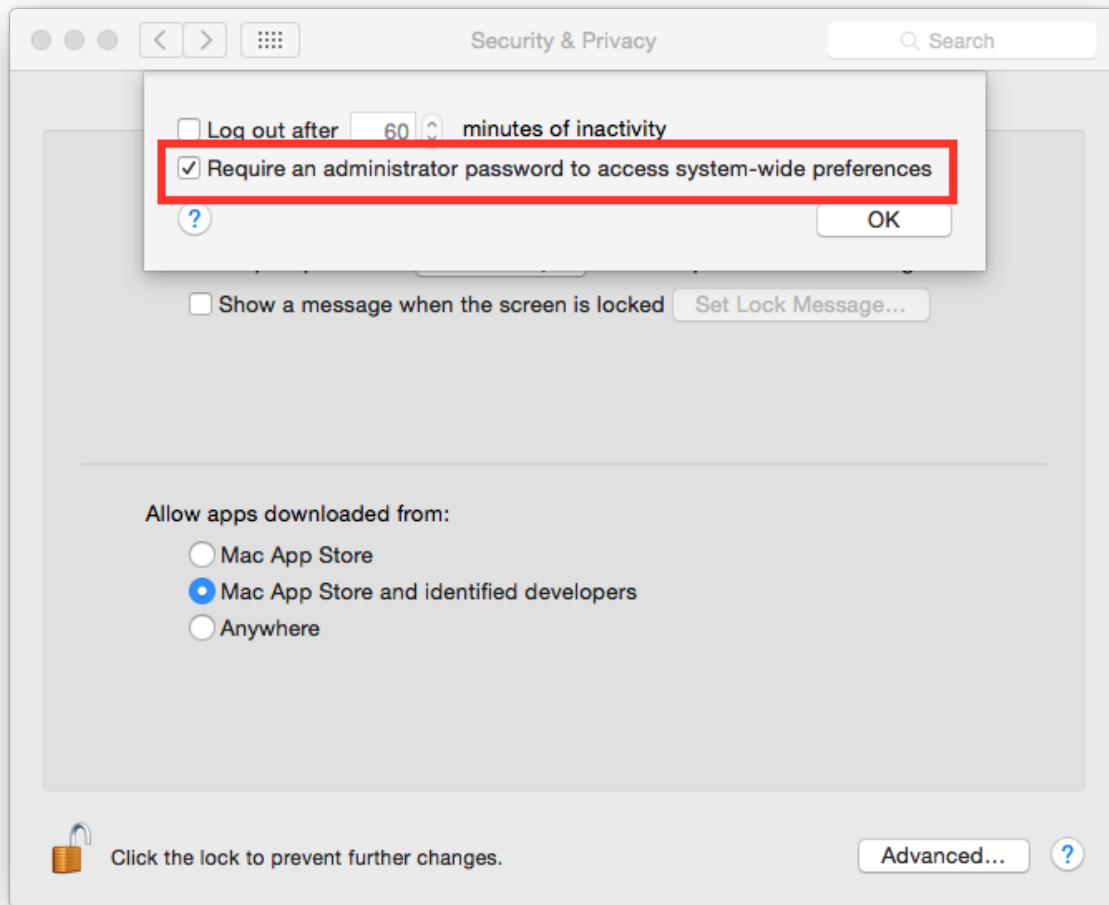
```
sudo pmset hibernatemode 25
```

### **Require an administration password**

Always require an administration password to access system settings. Go to:

System Preferences Security & Privacy Advanced

Check “Require an administrator password to access system-wide preferences”.

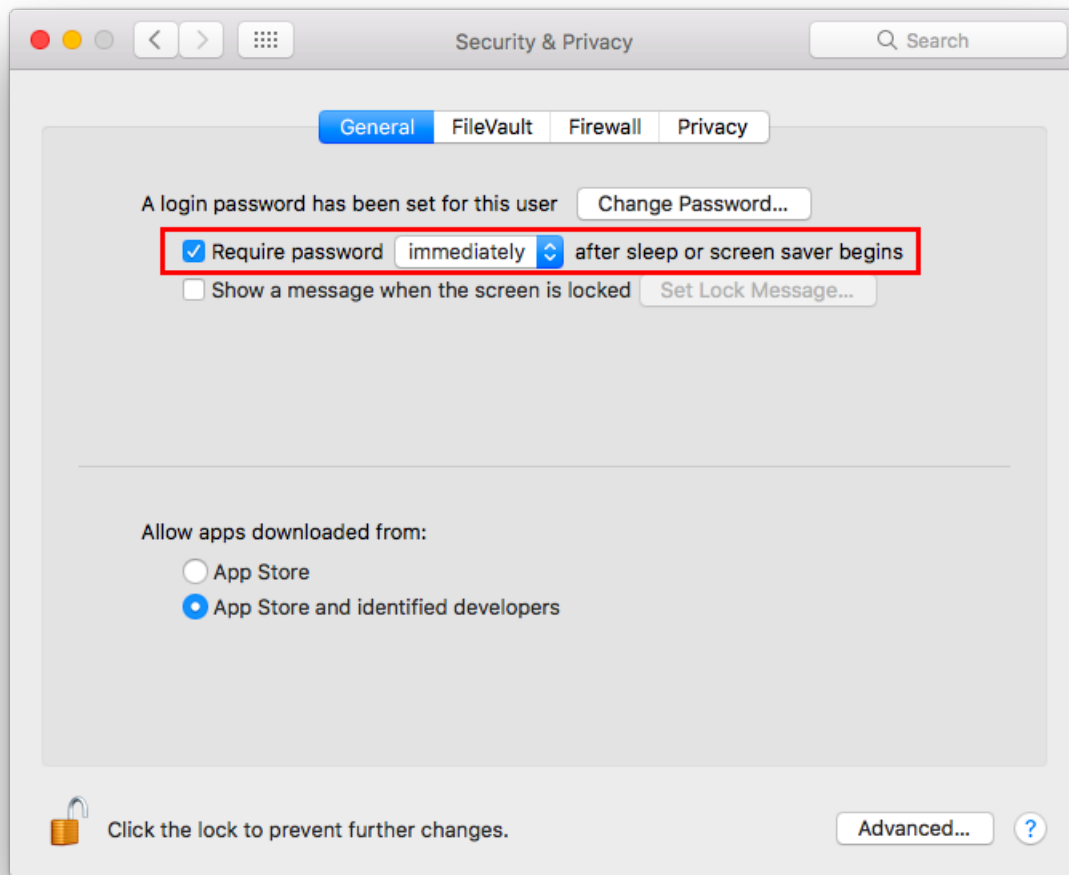


### Require password to un-lock

Requires password to un-lock from sleep or screen saver. Go to:

System Preferences Security & Privacy General

Set “Require password immediately after sleep or screen saver begins”.



### Save to Disk by Default

Many applications bundled in OS X, i.e. Text, save by default new documents to iCloud. It is suggested to set default save target to be a local disk, not iCloud with the following command, open Terminal and type:

```
defaults write NSGlobalDomain NSDocumentSaveNewDocumentsToCloud -bool false
```

### Set a Firmware Password

Enabling an optional firmware password offers an increased level of protection. A firmware password is set on the actual Mac logicboards firmware, it is an EFI password which prevents your Mac from being booted from an external boot volume, single user mode, or target disk mode, and it also prevents resetting of PRAM and the ability to boot into Safe Mode. Years ago firmware passwords could be easily bypassed by removing memory. These days Mac's firmware password isn't easily reset. Apple only suggests to bring your Mac in to an authorized Apple Service Provider and have them do it there.

It is suggested to set a firmware password:

- Power off your Mac and turn it on.

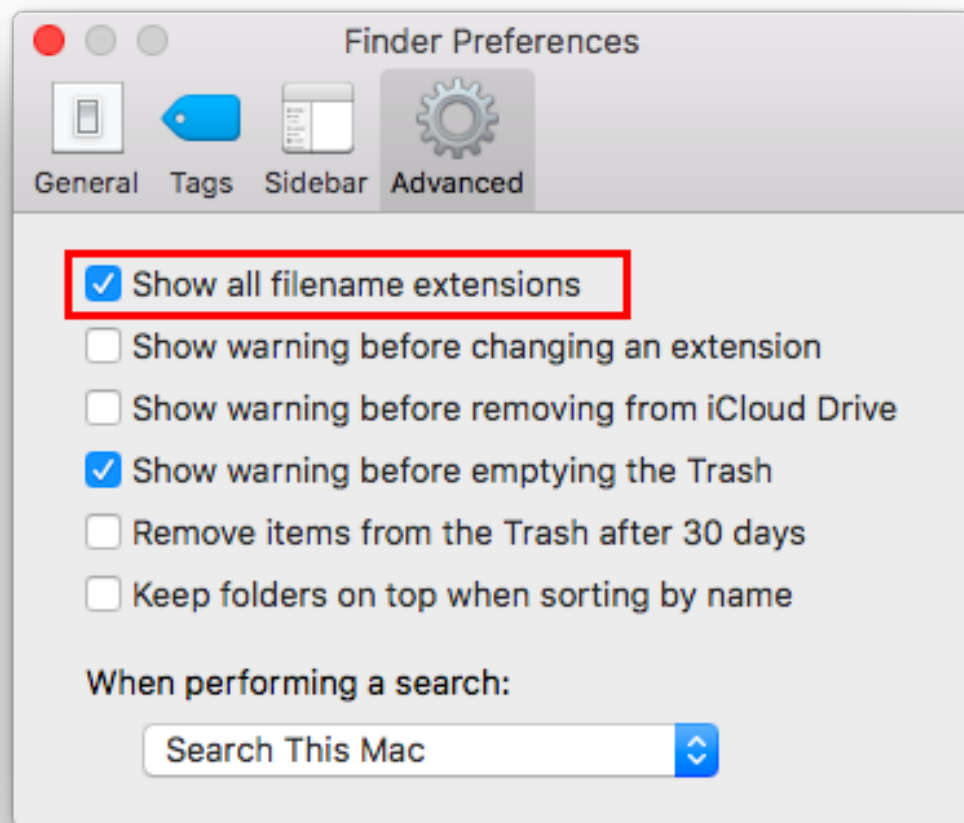
- Activate Recovery Mode (holding down the Command and R keys at boot).
- After a while OS X Utilities will appear.
- Click on the Utilities menu from the menu bar.
- Select Firmware Password Utility.
- Click on ‘Turn On Firmware Password’ and follow the wizard.
- When done, restart your Mac.

### Show all filename extensions

It is a good practice to always show file names extensions. Start Finder app. Go to:

Preferences Advanced

Check “Show all filename extensions”.

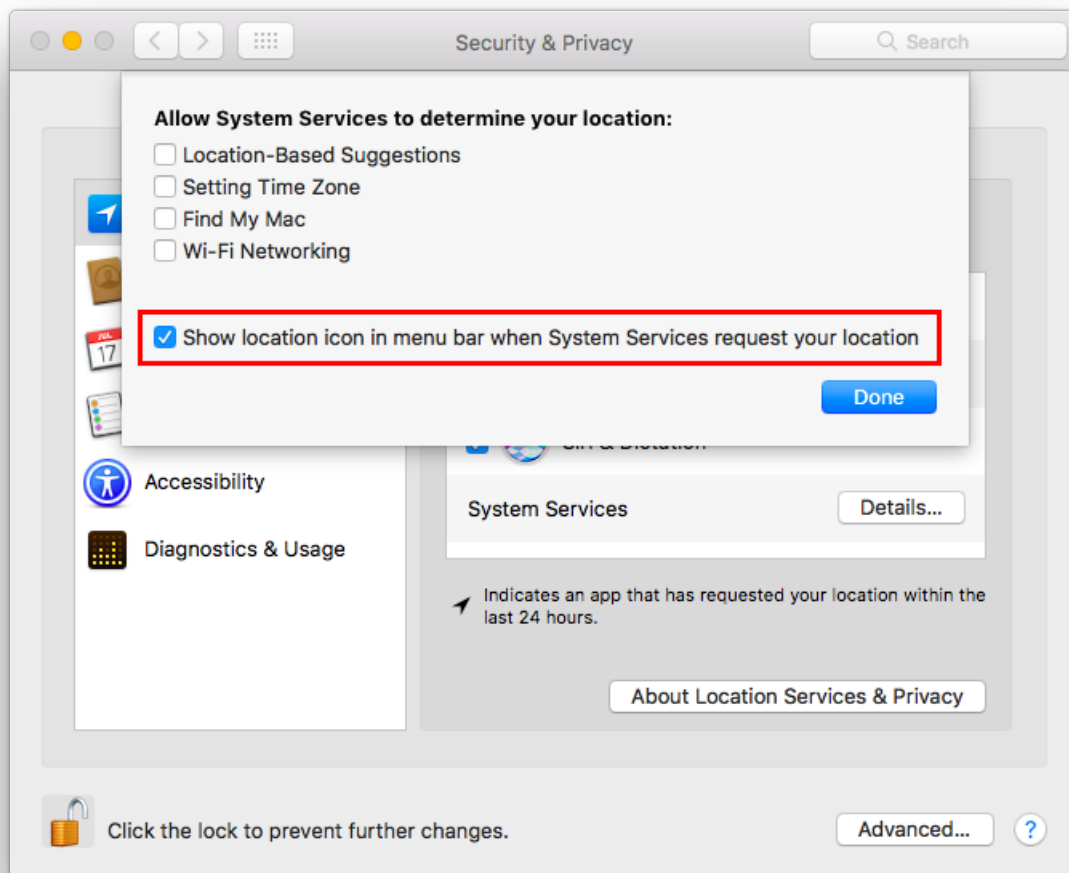


## Show when localization is used

System services could ask to use localization data. It is suggested to show location icon when localization data are requested. Go to:

System Preferences Security & Privacy Privacy Location Services

Select “System Services” and click “Details...”. Check “Show location icon in the menu bar when System Services request your location”.



## Users privilege separation

It is suggested to use different accounts for administration and normal use. Create an account with admin privileges for special tasks and maintenance and a regular user for your normal use. Don't use the same password for both.

## References

- <https://github.com/herrbischoff/awesome-osx-command-line>
- <http://www.frameless.org/2011/09/18/firewire-attacks-against-mac-os-lion-filevault-2-encryption/>

### 1.2.2 Mac OSX 10.10 Yosemite

According to [Wikipedia](#) Yosemite is “*OS X Yosemite (version 10.10) is the eleventh major release of OS X, Apple Inc.’s desktop and server operating system for Macintosh computers*”.

- *Applications*
- *Allow only signed apps*
- *Disable Diagnostics*
- *Disable Handoff*
- *Disable recent items*
- *Disable Spotlight localization*
- *Enable FileVault*
- *Enable Firewall*
- *Enable screen saver*
- *Empty trash securely*
- *Require an administration password*
- *Require password to un-lock*
- *Show all filename extensions*
- *Users privilege separation*

#### Applications

It is suggested to keep the /Applications/ directory as clean as possible, and having a separate directory for your personal apps lets you do that easily. Just create a folder named “Applications” in your home directory (or where you like).

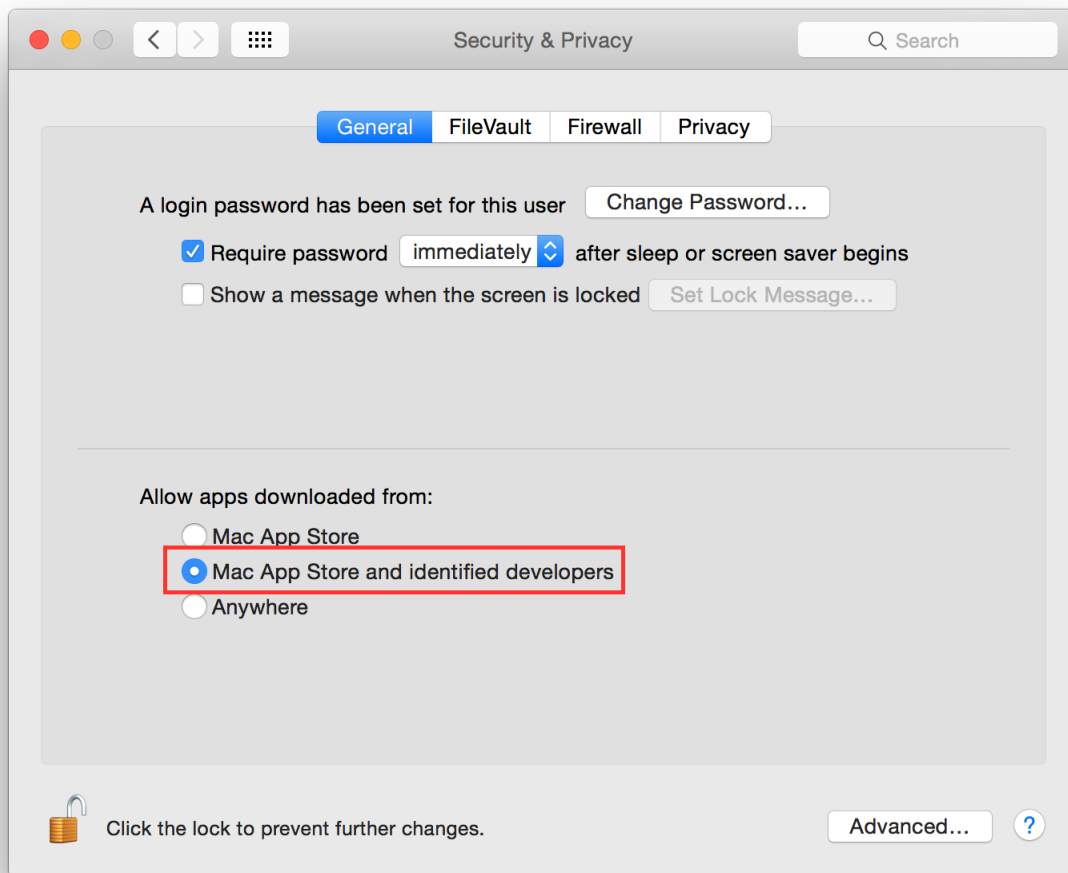
#### Allow only signed apps

It is suggested to never run untrusted code not signed with a proper key. To allow only apps signed by an authorized developer, go to:

System Preferences Security & Privacy General

Set “Allow apps download from” to “Mac App Store and identified developers”.



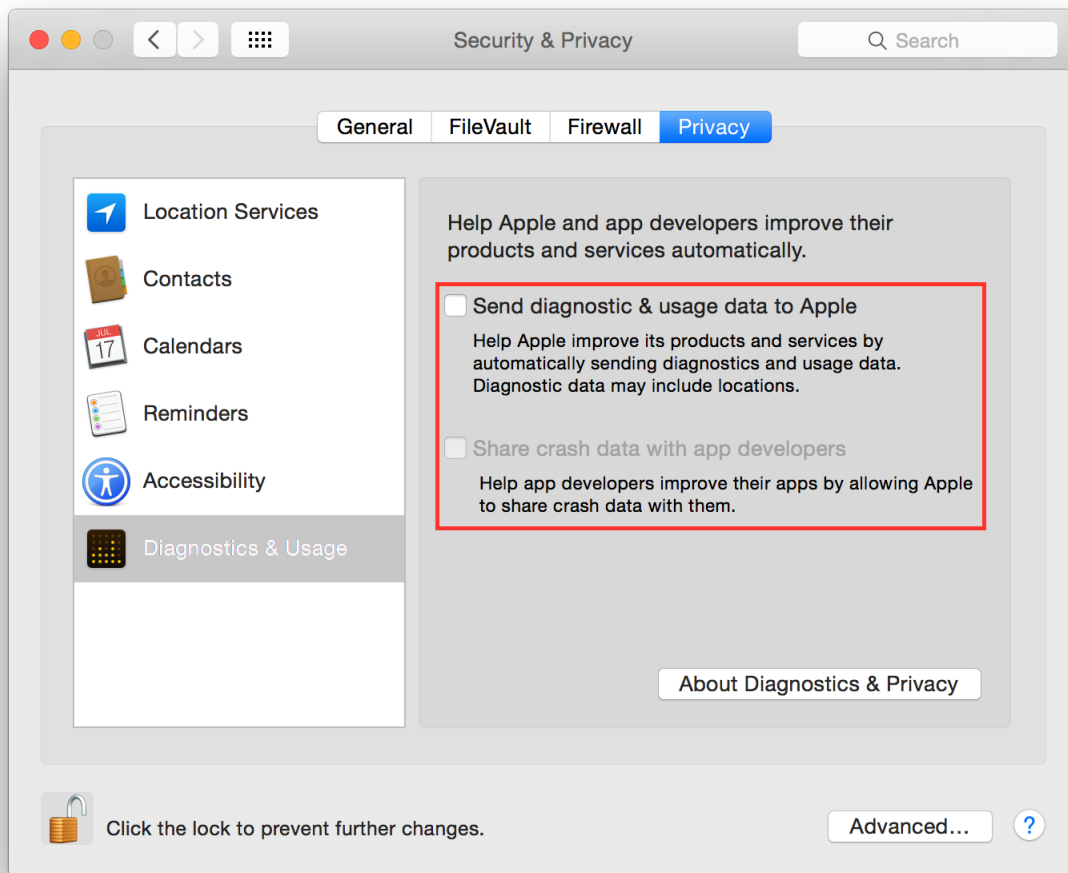


## Disable Diagnostics

It is suggested to disable diagnostic data and usage data sending to Apple. Go to:

System Preferences Security & Privacy Privacy Diagnostics & Usage

Uncheck “Send diagnostic & usage data to Apple”. Uncheck “Share crash data with app developers”.

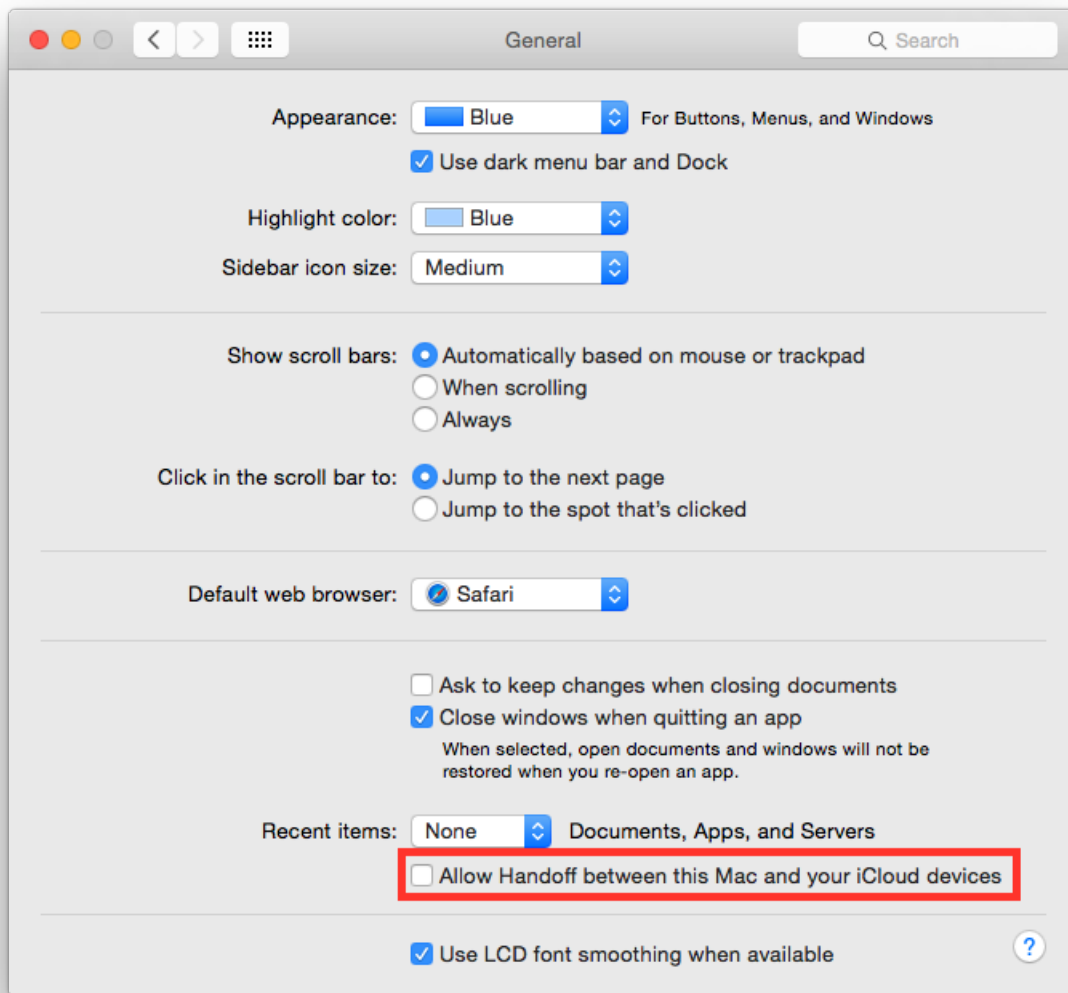


### Disable Handoff

Handoff is a great feature to keep your work in sync but it needs to send some data to Apple to work. It is suggested to disable it. Go to:

System Preferences General

Uncheck “Allow Handoff between this Mac and your iCloud devices”.

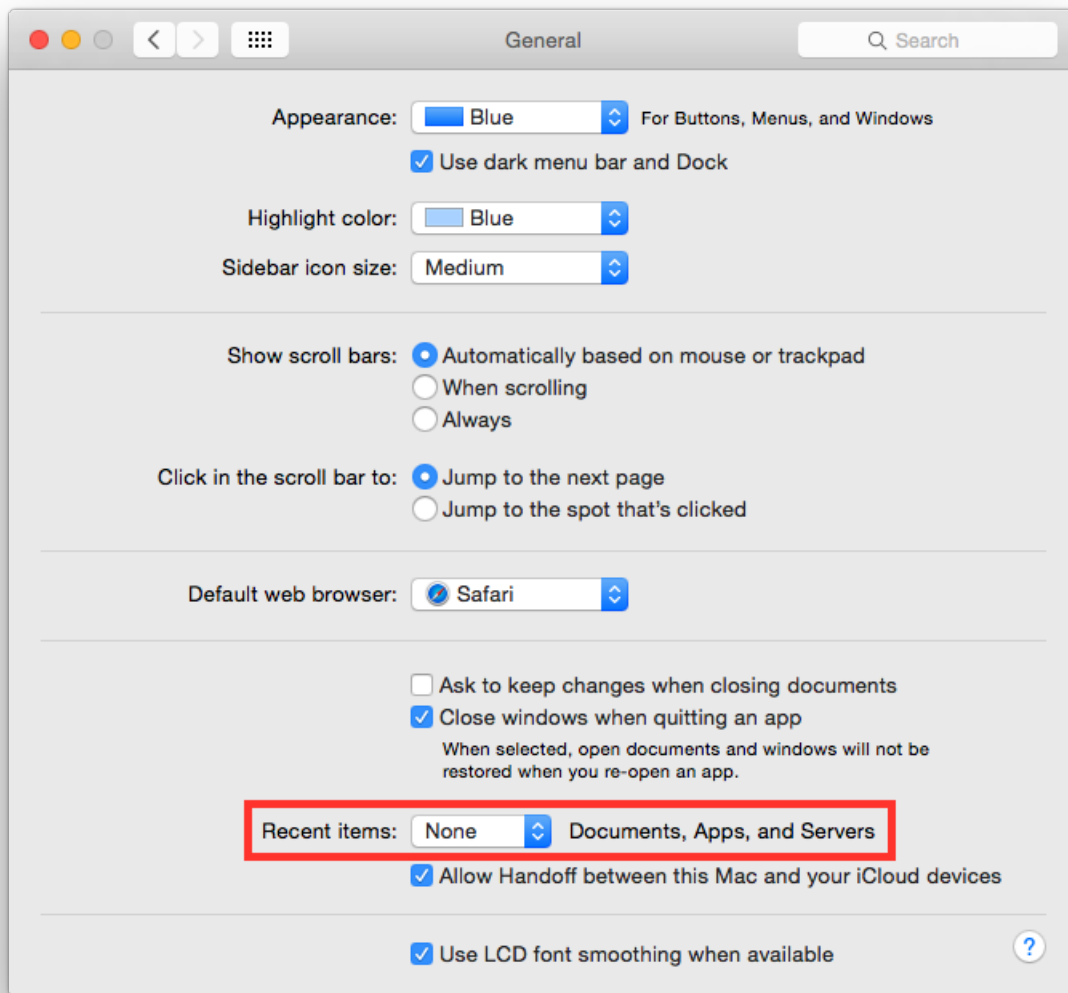


### Disable recent items

Do not track last recently used items. Go to:

System Preferences General

Set “Recent items” to “None”.

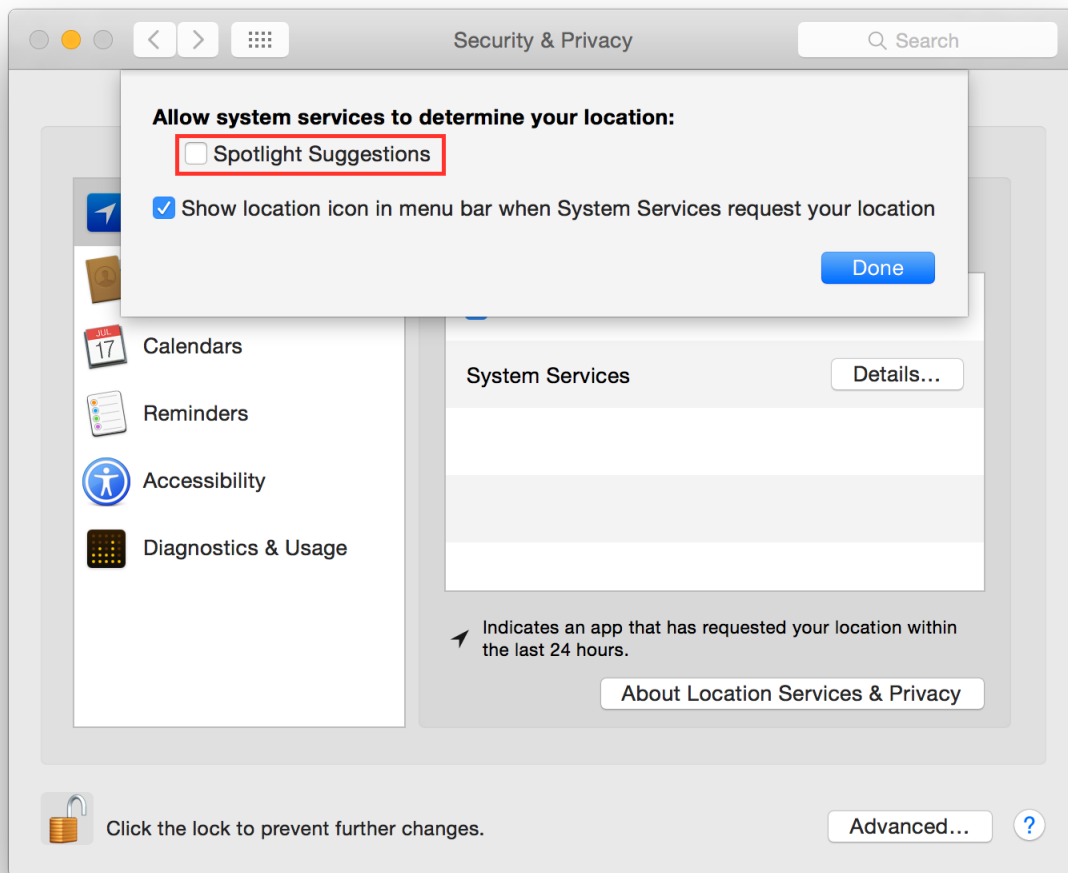


## Disable Spotlight localization

By default Spotlight is allowed to use localization services to help you offering localized results. Go to:

System Preferences Security & Privacy Privacy Location Services

Select “System Services” and click “Details...”. Uncheck “Spotlight Suggestions”.



## Enable FileVault

It is suggested to enable FileVault to enable full disk encryption on your device. It should be already enabled by default. Go to:

System Preferences Security & Privacy FileVault

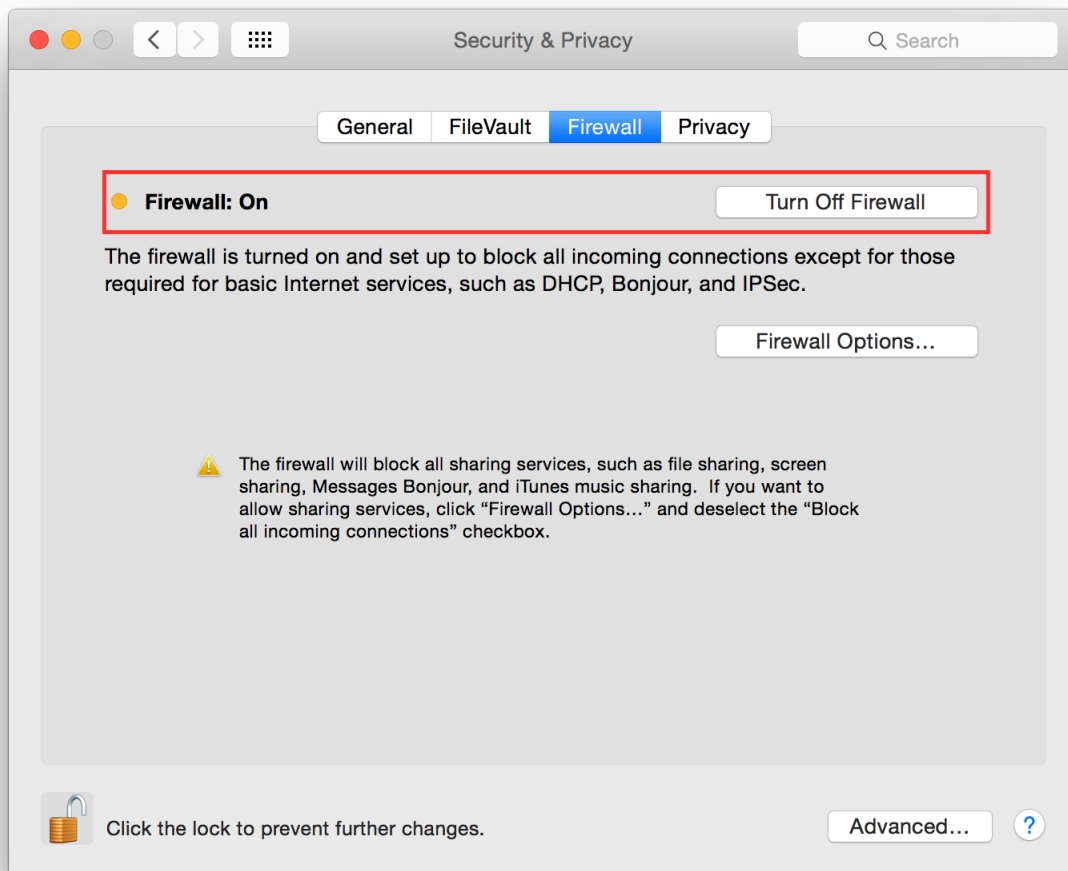
Enable FileVault.

## Enable Firewall

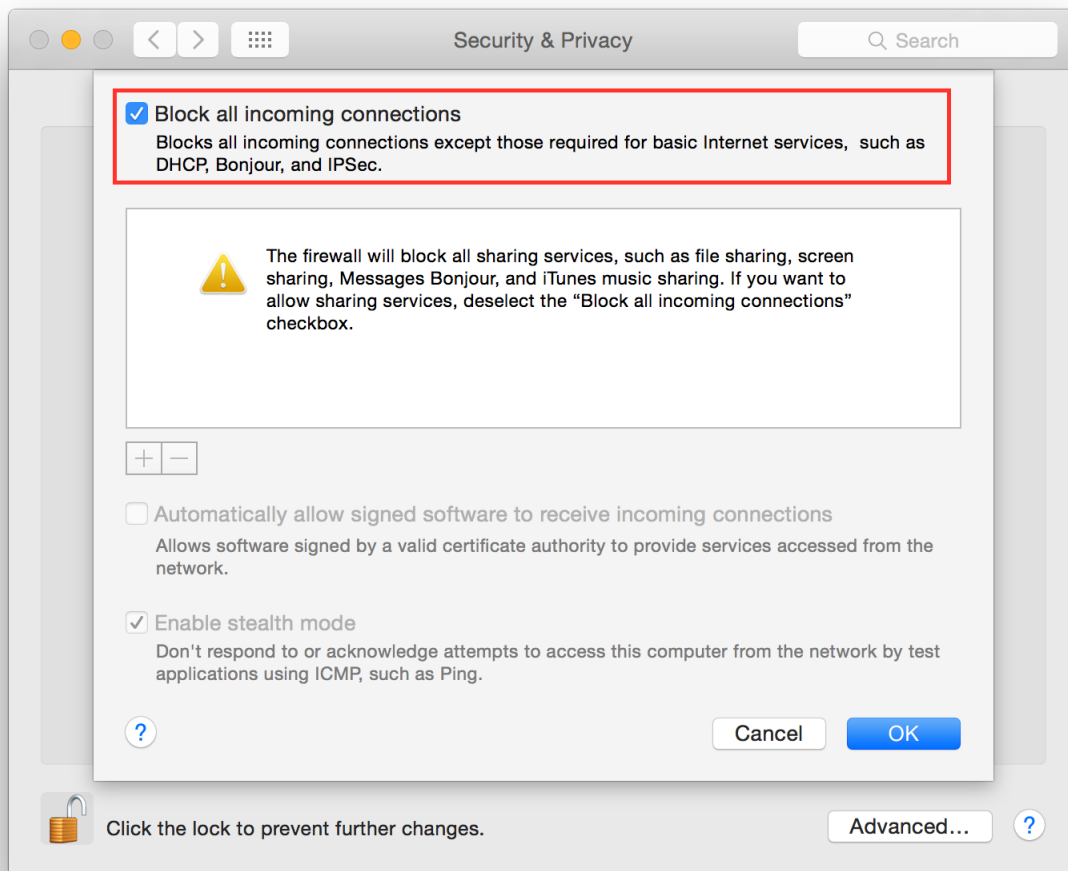
It is suggested to enable the Firewall and always have one running. Go to:

System Preferences Security & Privacy Firewall

Click on "Turn On Firewall".



Now click on “Firewall options”, a new panel will appear. Click on “Block all incoming connections”.

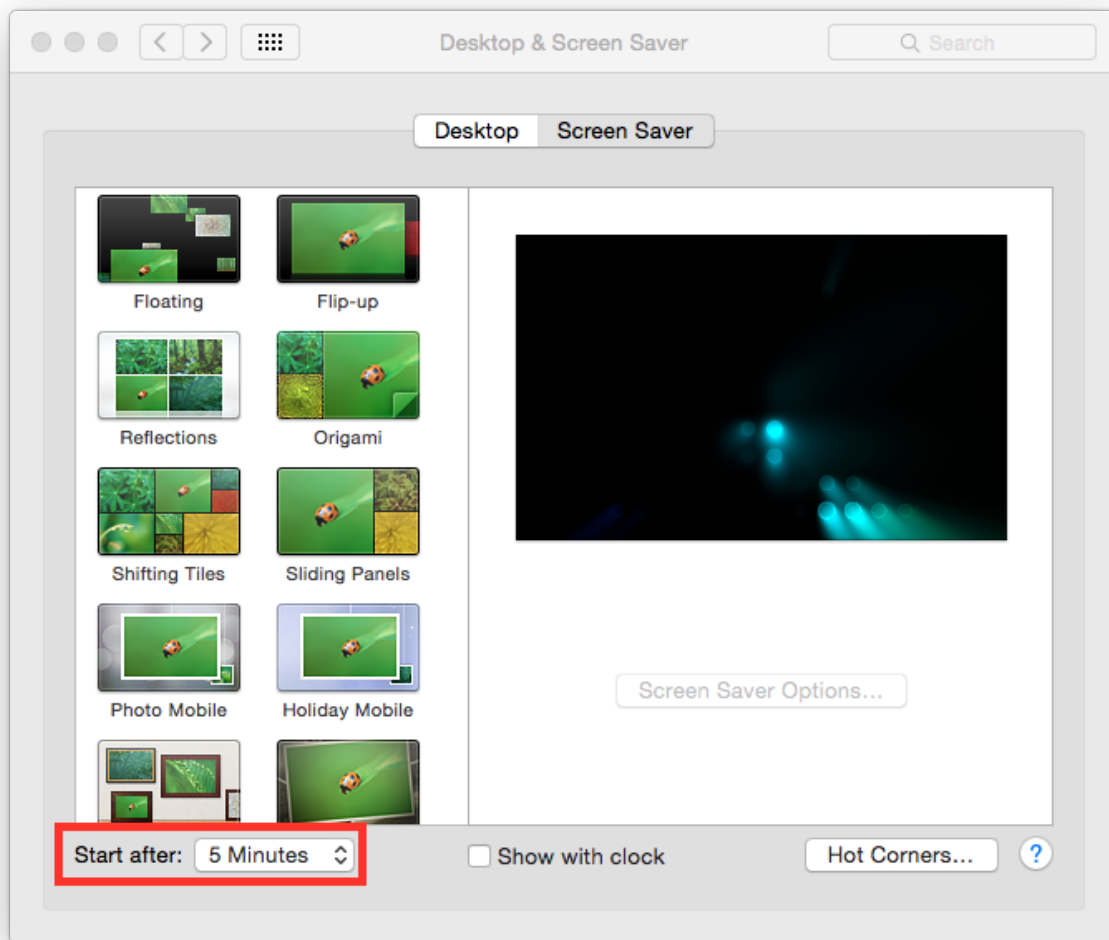


### Enable screen saver

It is suggested to enable the screen saver to automatically lock your screen after a while. Go to:

System Preferences Desktop & Screen Saver Screen Saver

Set “Start after” to “5 Minutes”.



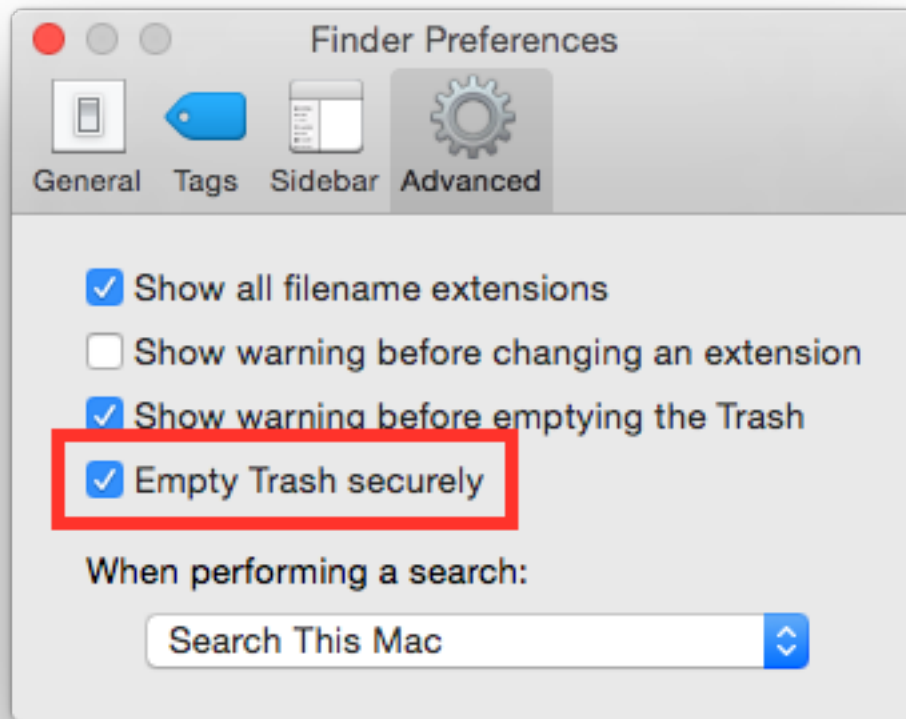
## Empty trash securely

It is a good practice to always empty your trash securely. Start Finder app. Go to:

Preferences Advanced

Check “Empty trash securely”.



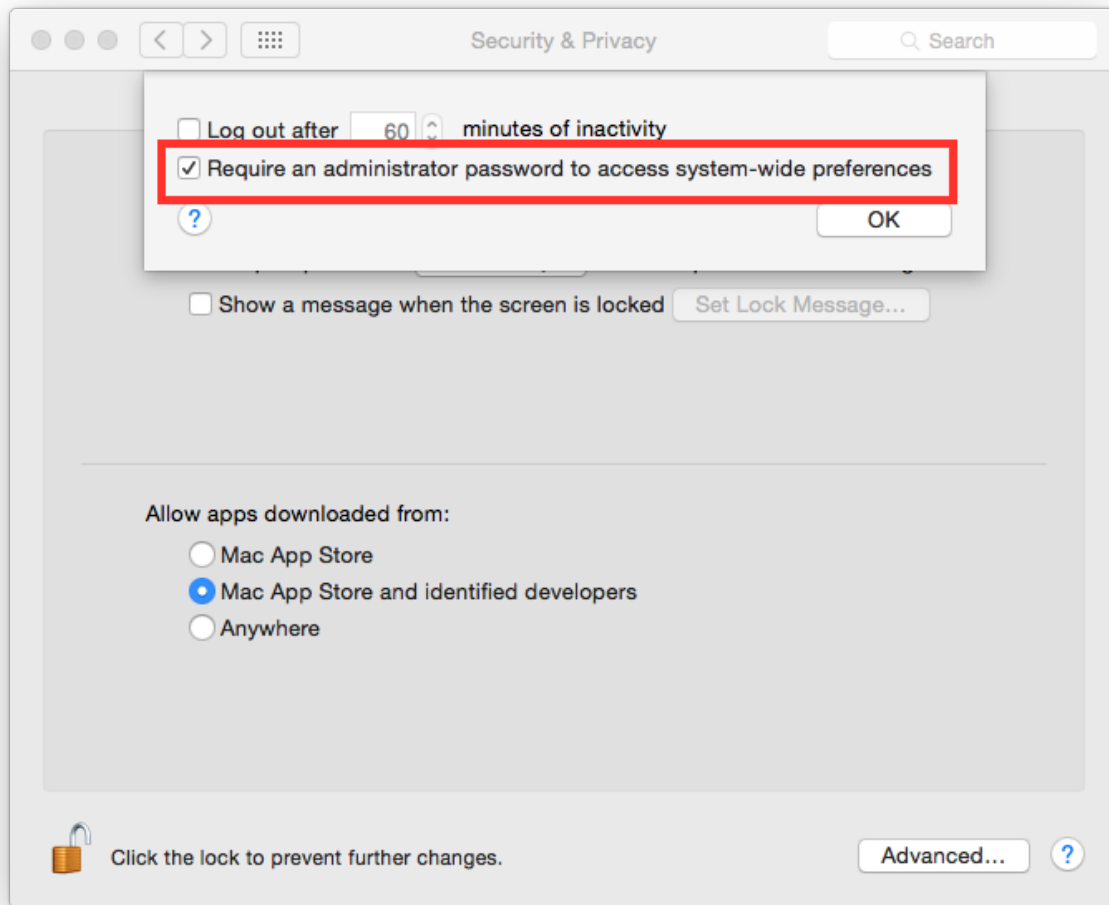


### Require an administration password

Always require an administration password to access system settings. Go to:

System Preferences Security & Privacy Advanced

Check “Require an administrator password to access system-wide preferences”.

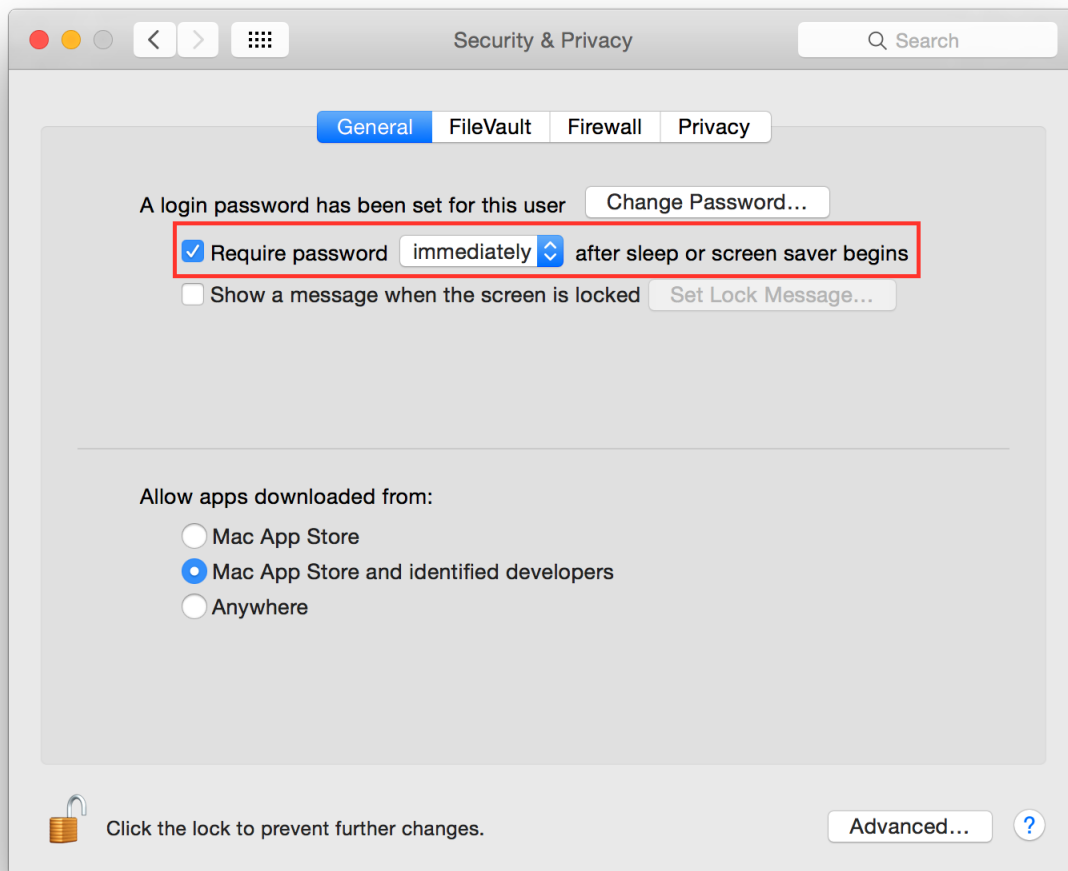


### Require password to un-lock

Requires password to un-lock from sleep or screen saver. Go to:

System Preferences Security & Privacy General

Set “Require password immediately after sleep or screen saver begins”.

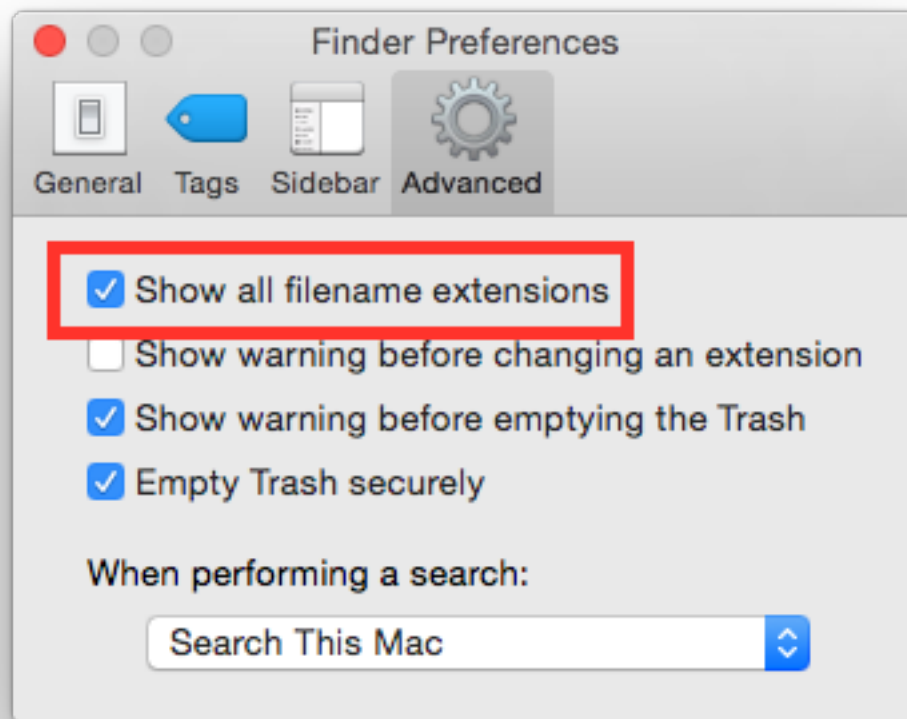


### Show all filename extensions

It is a good practice to always show file names extensions. Start Finder app. Go to:

Preferences Advanced

Check “Show all filename extensions”.



### Users privilege separation

It is suggested to use different accounts for administration and daily activities. Create an account with admin privileges for special tasks and maintenance and a normal user for your daily use.

### 1.2.3 Mac OSX 10.11 El Capitan

According to [Wikipedia](#) El Capitan is “(version 10.11) is the twelfth major release of OS X, Apple Inc.’s desktop and server operating system for Macintosh computers. It is the successor to OS X Yosemite and focuses mainly on performance, stability and security. Following the California landmark-based naming scheme introduced with OS X Mavericks, El Capitan was named after a rock formation in Yosemite National Park.”.

- *Applications*
- *Allow only signed apps*
- *Check Privacy permissions*
- *Destroy FileVault Keys*

- *Disable Bonjour*
- *Disable Creation of Metadata Files*
- *Disable Diagnostics*
- *Disable Guest user*
- *Disable Handoff*
- *Disable password hints*
- *Disable recent items*
- *Disable Spotlight localization*
- *Disable Spotlight Suggestions*
- *Enable FileVault*
- *Enable Firewall*
- *Enable screen saver*
- *Empty trash securely*
- *Erase free space*
- *Power off memory during standby*
- *Require an administration password*
- *Require password to un-lock*
- *Save to Disk by Default*
- *Set a Firmware Password*
- *Show all filename extensions*
- *Show when localization is used*
- *Users privilege separation*
- *References*

## Applications

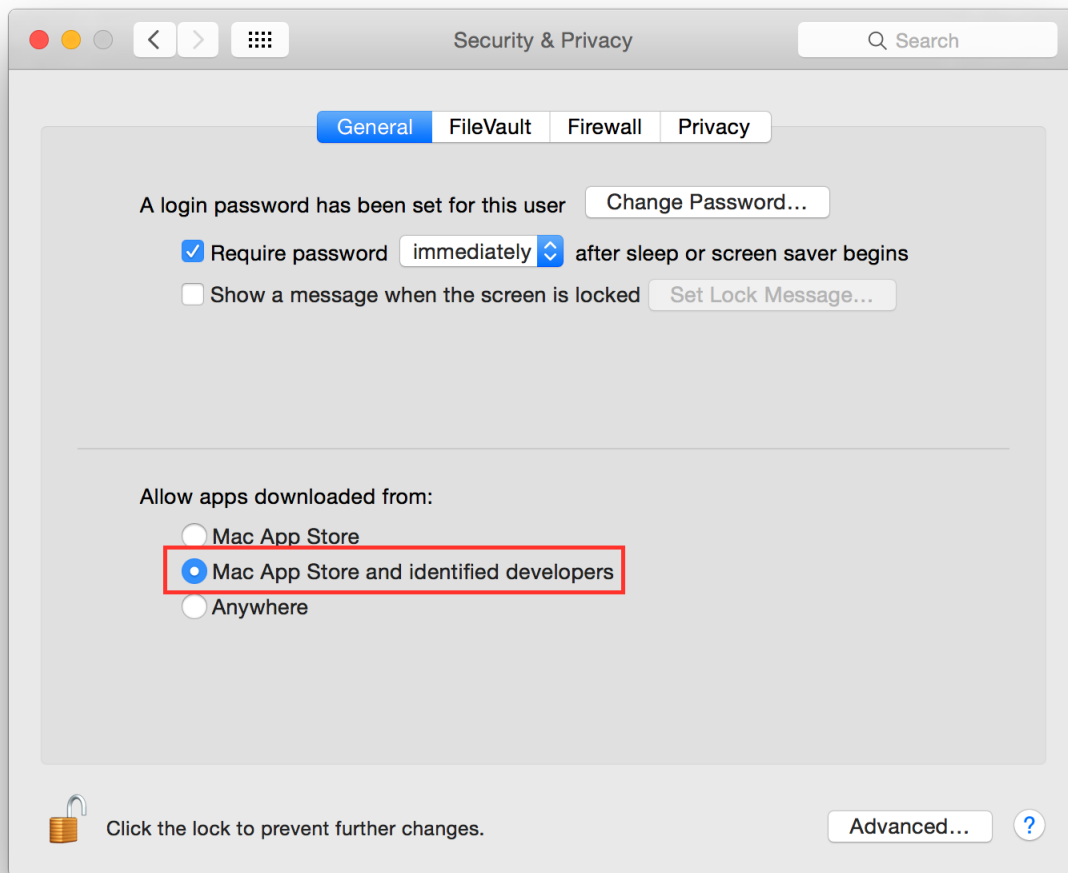
It is suggested to keep the `/Applications/` directory as clean as possible, having a separate directory for your personal apps lets you do that easily. Just create a folder named “Applications” in your home directory (or where you like) and install all applications there. Apps installed via App Store or some special apps cannot live in a custom Applications folder, so you have to keep them in the original path.

### Allow only signed apps

It is suggested to never run untrusted code not signed with a proper key. To allow only apps signed by an authorized developer, go to:

System Preferences Security & Privacy General

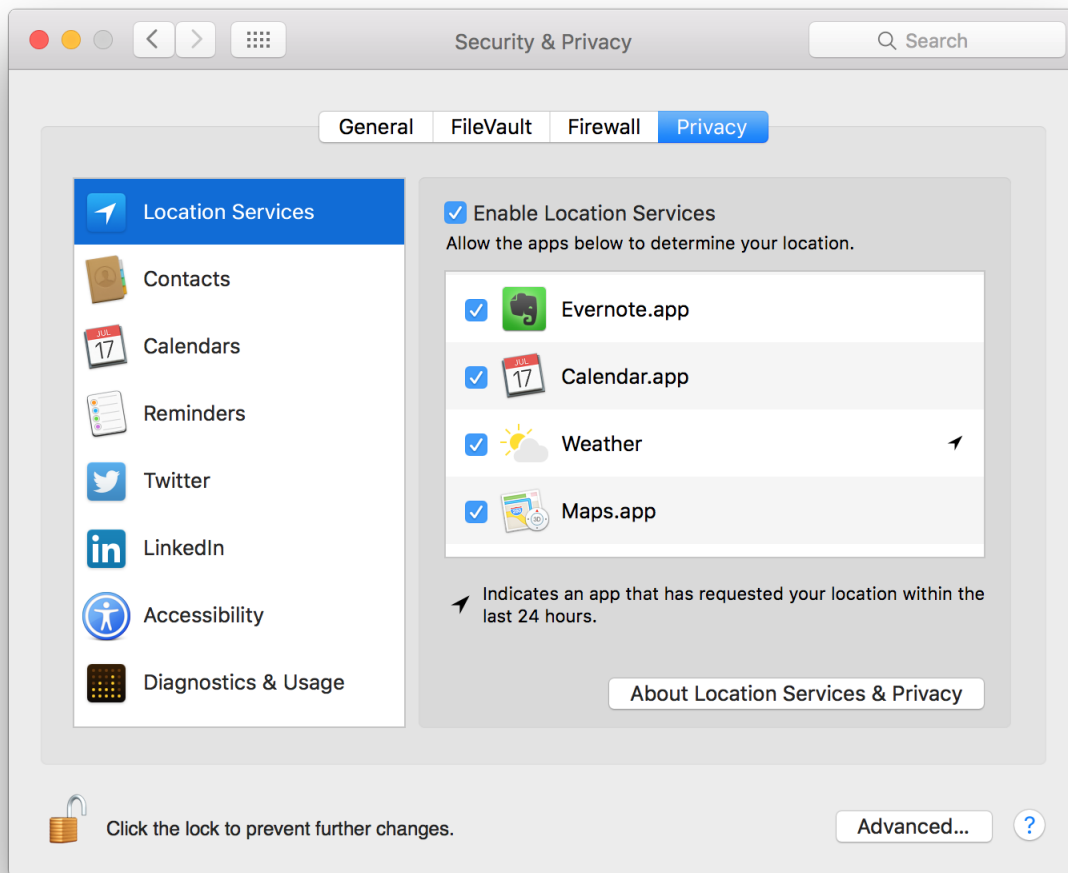
Set “Allow apps download from” to “Mac App Store and identified developers” or if you want to be more strict and you install applications only via App Store set it to “Mac App Store”.



### Check Privacy permissions

OS X allows you to track all applications requesting access to some sort of sensitive data, for example your location or your contacts. It is suggested to periodically check the list of applications requesting access to sensitive data and review their permissions. To show the list of these applications go to:

System Preferences Security & Privacy Privacy



## Destroy FileVault Keys

By default File Vault keys are kept when system goes in standby mode. As suggested by *man pmset*:

**destroyfvkeyonstandby - Destroy File Vault Key when going to standby mode.** By default File vault keys are retained even when system goes to standby. If the keys are destroyed, user will be prompted to enter the password while coming out of standby mode.(value: 1 - Destroy, 0 - Retain)

It is suggested to configure your system to destroy File Vault keys when entering in standby mode with the following command:

```
sudo pmset destroyfvkeyonstandby 1
```

## Disable Bonjour

According to [Wikipedia](#) Bonjour is “Apple’s implementation of Zero-configuration networking (Zeroconf), a group of technologies that includes service discovery, address assignment, and hostname resolution. Bonjour locates devices such as printers, other computers, and the services that those devices offer on a local network using multicast Domain Name System (mDNS) service records”.

Bonjour sends some data about your OS on the network, so in some cases, in a not trusted network you would like to disable it.

To disable Bonjour use the following command in a Terminal:

```
sudo defaults write /System/Library/LaunchDaemons/com.apple.mDNSResponder_  
↳ProgramArguments -array-add "-NoMulticastAdvertisements"
```

To enable Bonjour use the following command in a Terminal:

```
sudo defaults write /System/Library/LaunchDaemons/com.apple.mDNSResponder_  
↳ProgramArguments -array "/usr/sbin/mDNSResponder" "-launchd"
```

### Disable Creation of Metadata Files

By default OS X creates metadata files in each directory to speed up browsing. These files could leak metadata, it is suggested to avoid creation of .DS\_Store and AppleDouble files.

Disable Creation of Metadata Files on Network Volumes with the following command in a Terminal:

```
defaults write com.apple.desktopservices DSDontWriteNetworkStores -bool true
```

Disable Creation of Metadata Files on USB Volumes with the following command in a Terminal:

```
defaults write com.apple.desktopservices DSDontWriteUSBStores -bool true
```

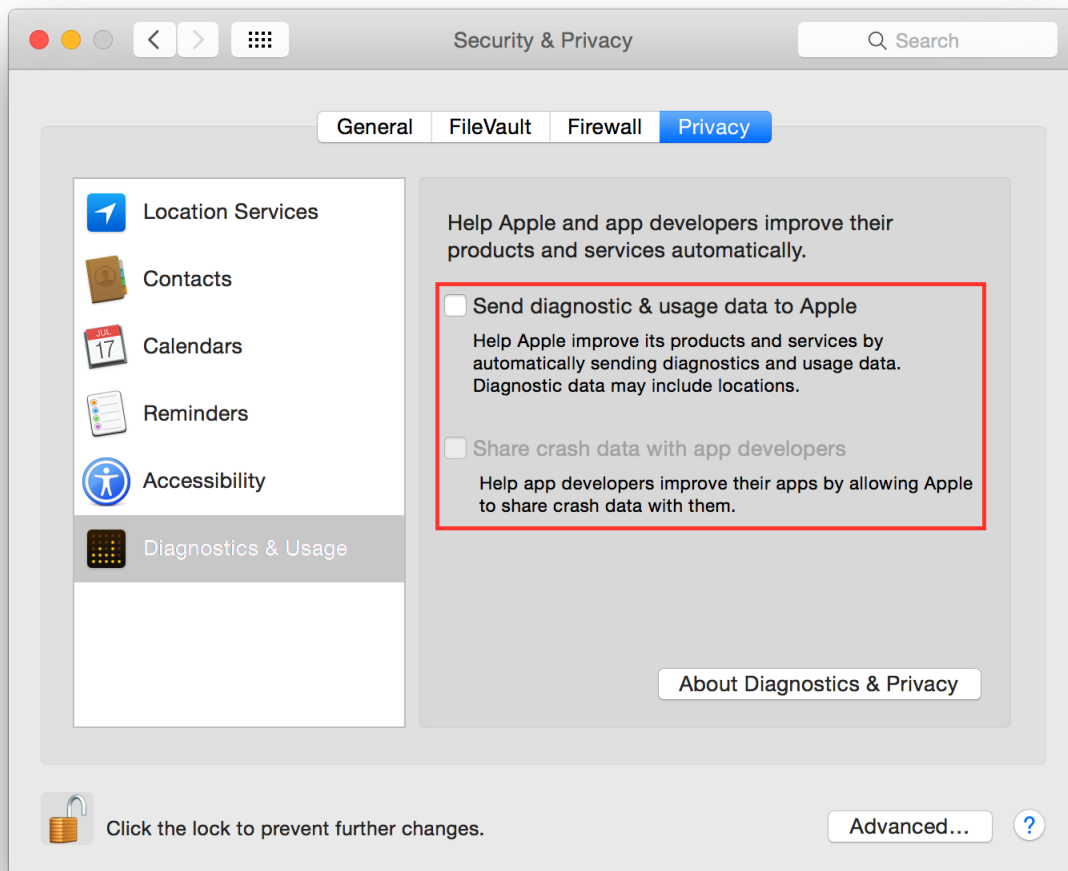
### Disable Diagnostics

It is suggested to disable diagnostic data and usage data sending to Apple. Go to:

System Preferences Security & Privacy Privacy Diagnostics & Usage

Un-check “Send diagnostic & usage data to Apple”. Un-check “Share crash data with app developers”.



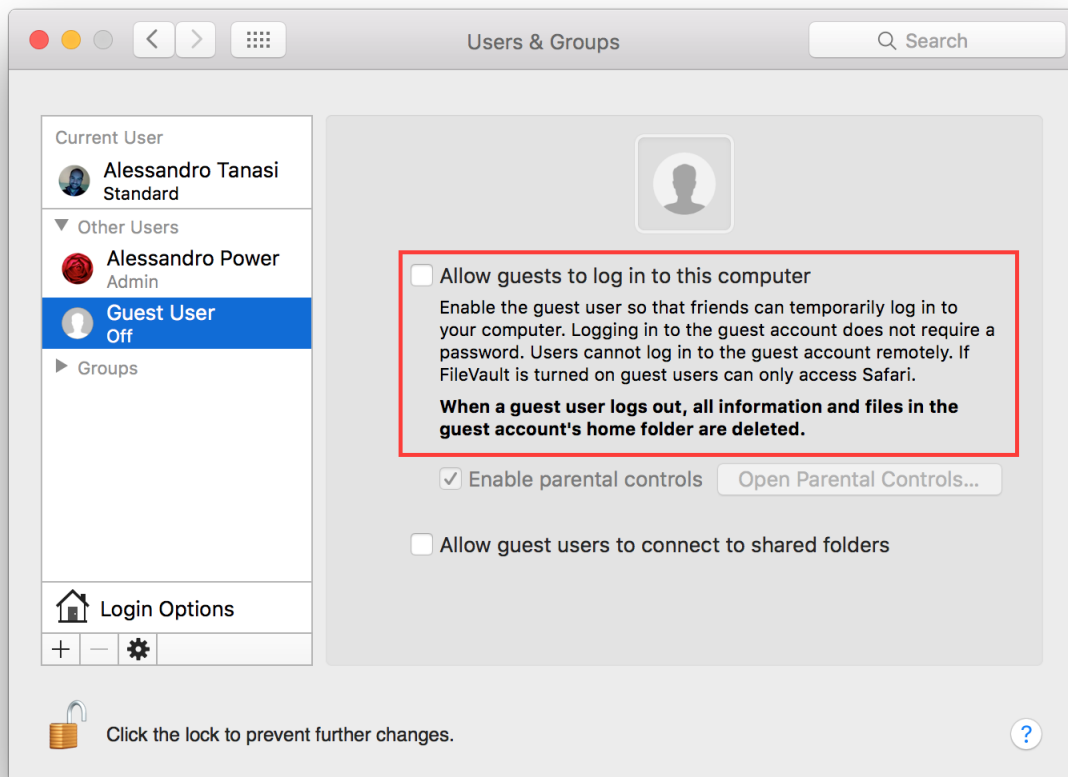


## Disable Guest user

Mac OS X comes with a *Guest* user enabled by default, it permits the use of your device in a restricted environment to anyone. It is suggested to disable the *Guest* user, go to:

System Preferences Users & Groups Guest User

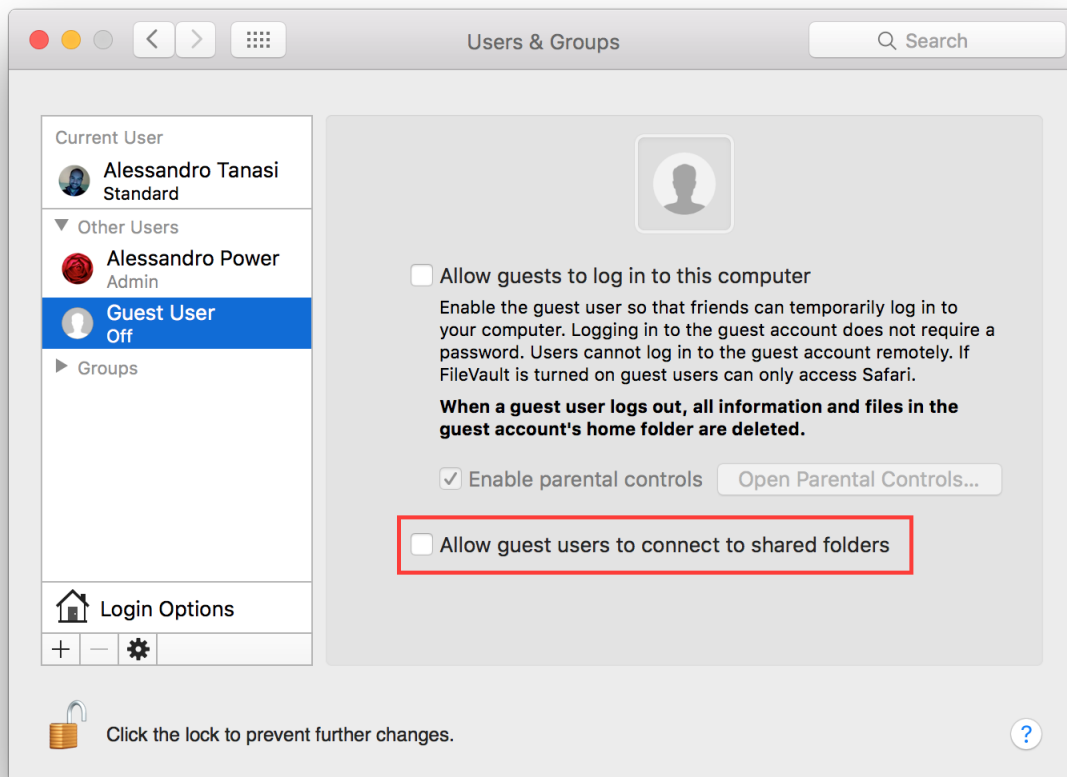
Un-check “Allow guests to log in to this computer”.



It is suggested to disable guest access to shared folders, if you are not using it, go to:

System Preferences Users & Groups Guest User

Un-check “Allow guest users to connect to shared folders”.

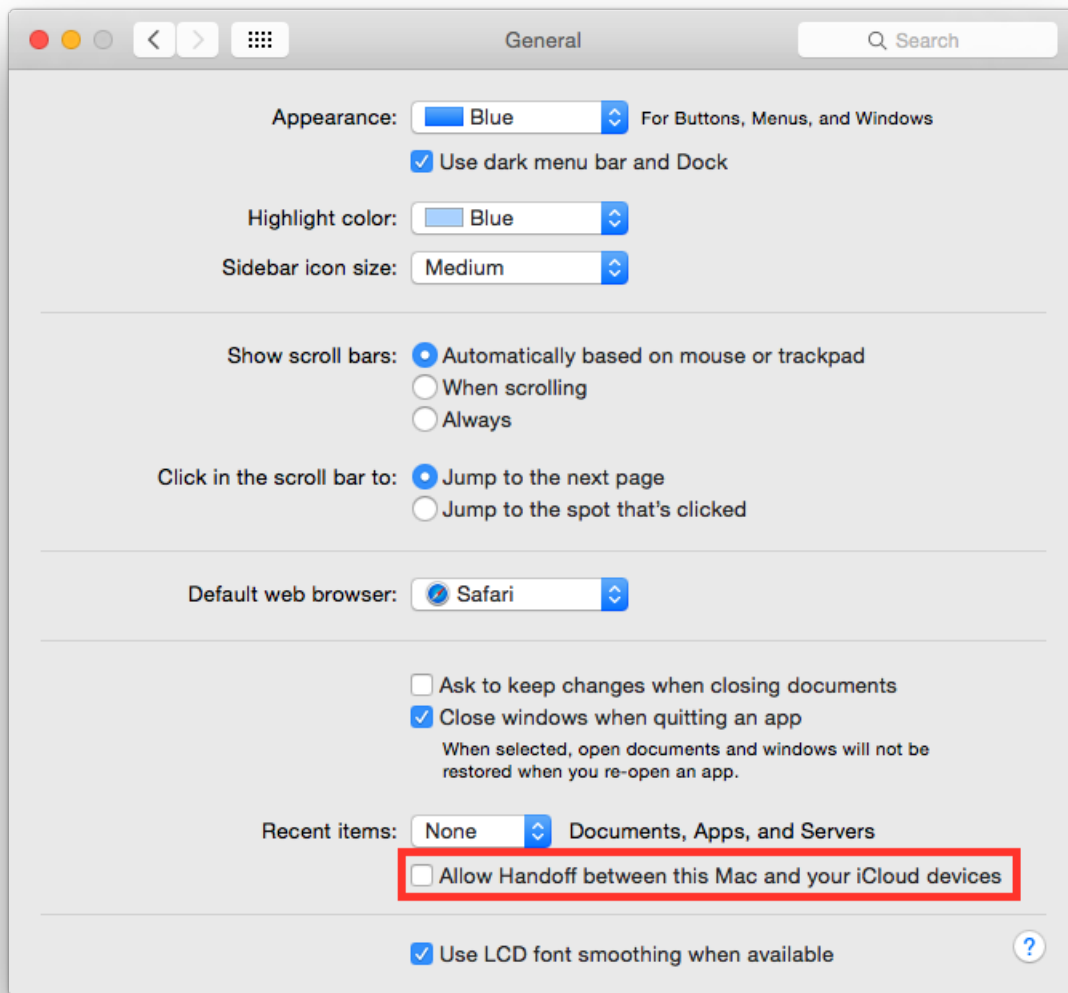


## Disable Handoff

Handoff is a great feature to keep your work in sync between Apple devices. Due to its implementation it needs to send some data to Apple iCloud to work, so in some way it is leaking your data. It is suggested to disable it. Go to:

System Preferences General

Un-check "Allow Handoff between this Mac and your iCloud devices".

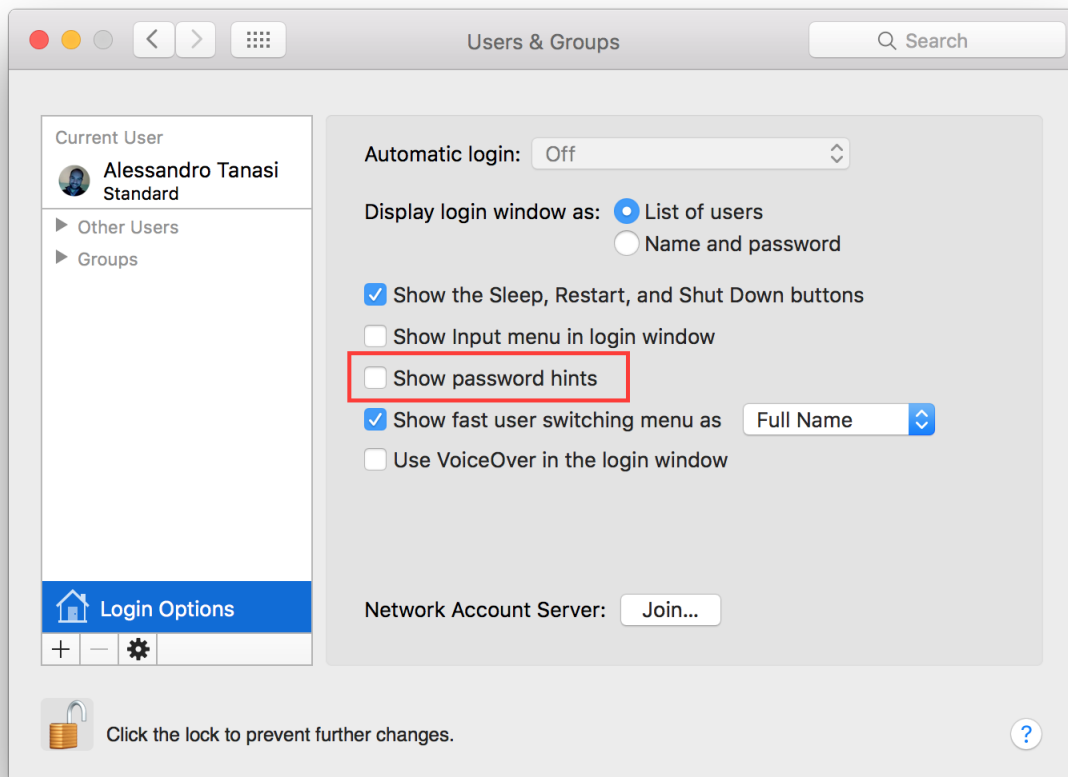


## Disable password hints

Passwords hints are supposed to help an user to remember his password but could also help attackers. It is suggested to disable password hints, go to:

System Preferences Users & Groups Login Options

Un-check “Show password hints”.

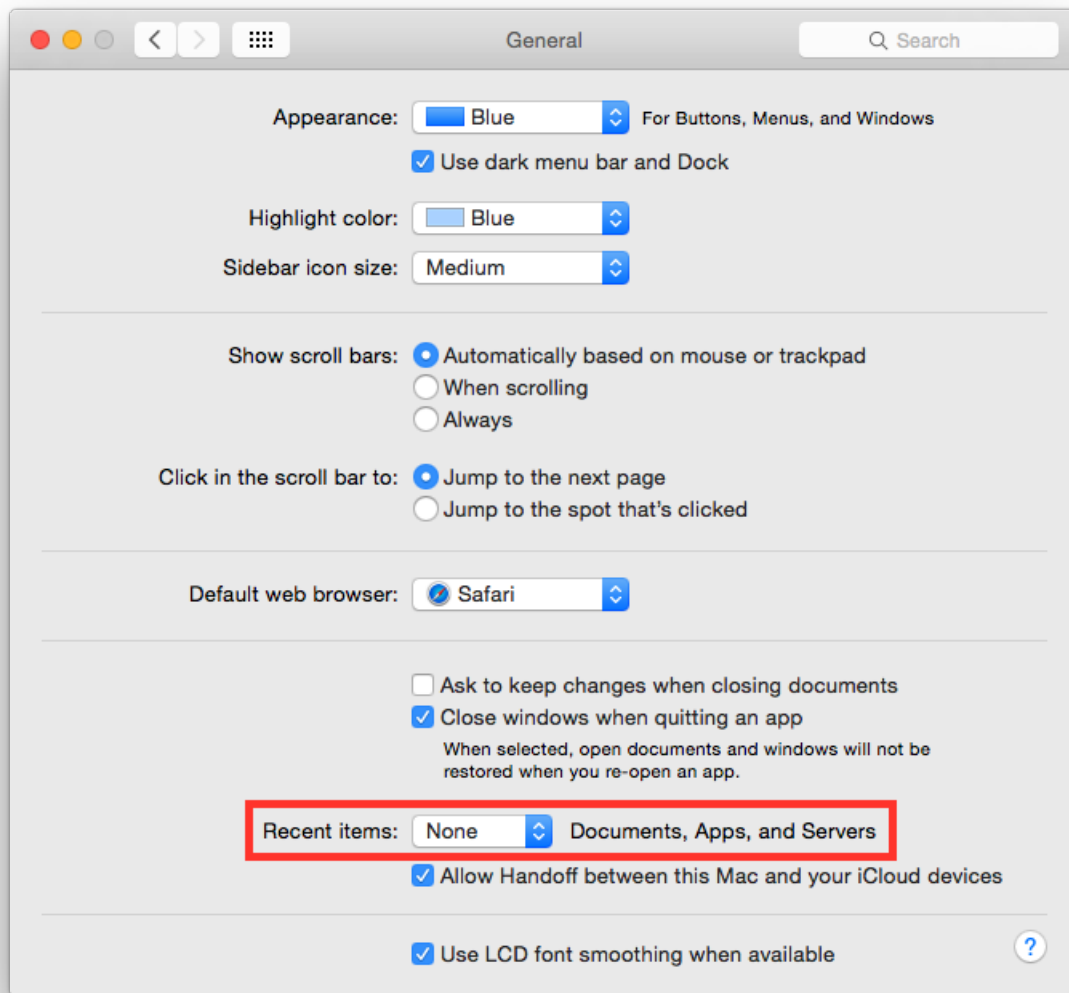


## Disable recent items

Recent items are used to track your latest activity, it is also a feature used in forensics investigation to create the user activity timeline. It is suggested to not track last recently used items. Go to:

System Preferences General

Set "Recent items" to "None".

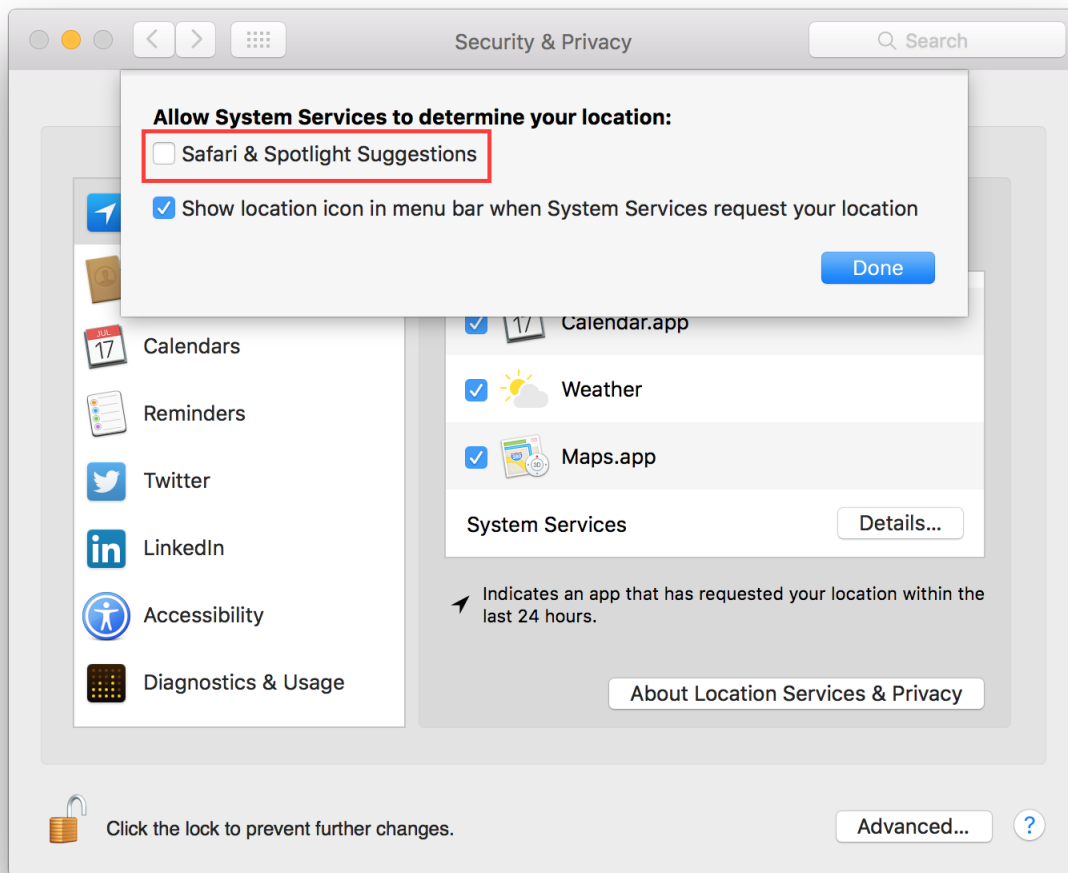


## Disable Spotlight localization

By default Spotlight is allowed to use localization services to help you offering localized results. Due to his implementation it needs to send your position to a remote service. It is suggested to disable this behavior. Go to:

System Preferences Security & Privacy Privacy Location Services

Select “System Services” and click “Details...”. Un-check “Safari & Spotlight Suggestions”.

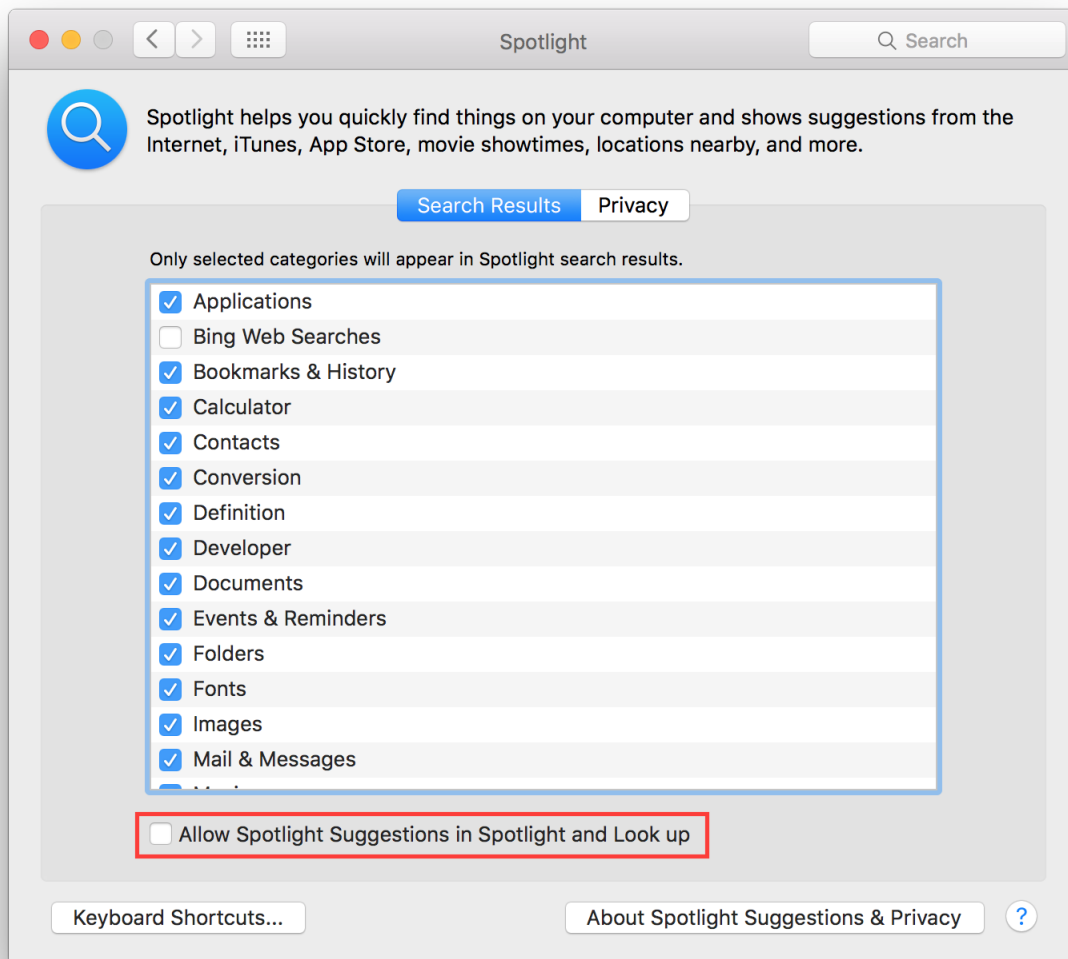


## Disable Spotlight Suggestions

By default Spotlight shows suggestions from the Internet, it sends your search to Apple services and provides results back. It is suggested to use Spotlight only locally to prevent leaking your search. To disable Spotlight Suggestions go to:

System Preferences Spotlight

Un-check “Allow Spotlight Suggestions in Spotlight and Look Up”.

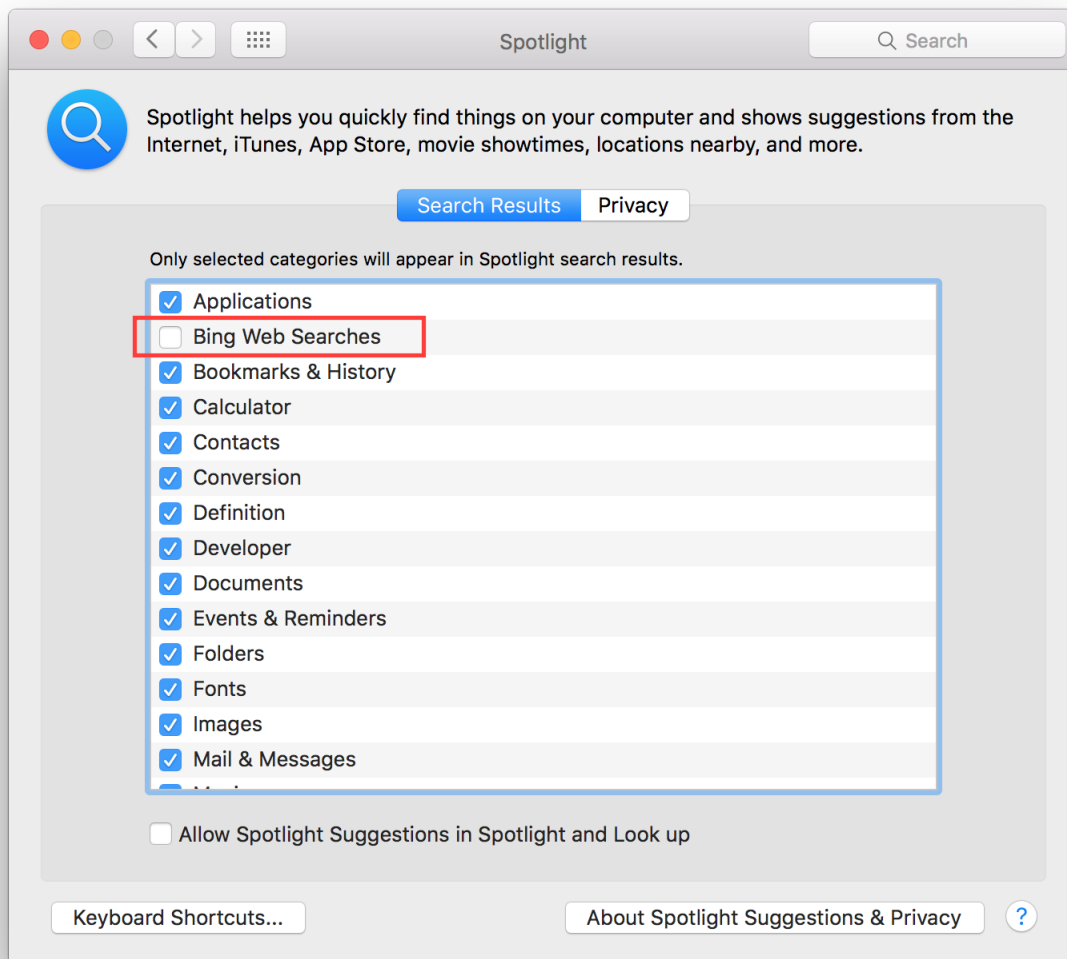


It is suggested to disable results from Bing to avoid leaking your search to Bing, go to:

System Preferences Spotlight

Un-check “Bing Web Searches” from the list of results categories.





## Enable FileVault

It is suggested to enable FileVault to enable full disk encryption on your device. It should be already enabled by default. Go to:

System Preferences Security & Privacy FileVault

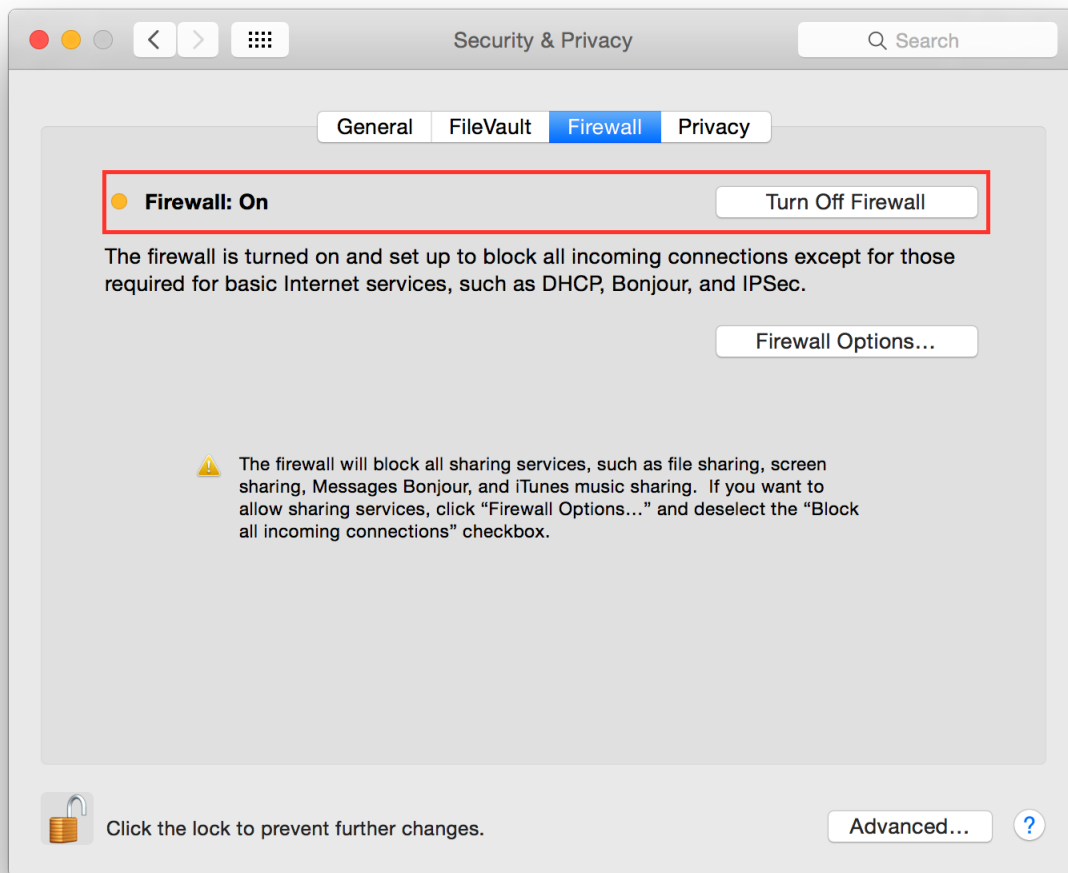
Enable FileVault.

## Enable Firewall

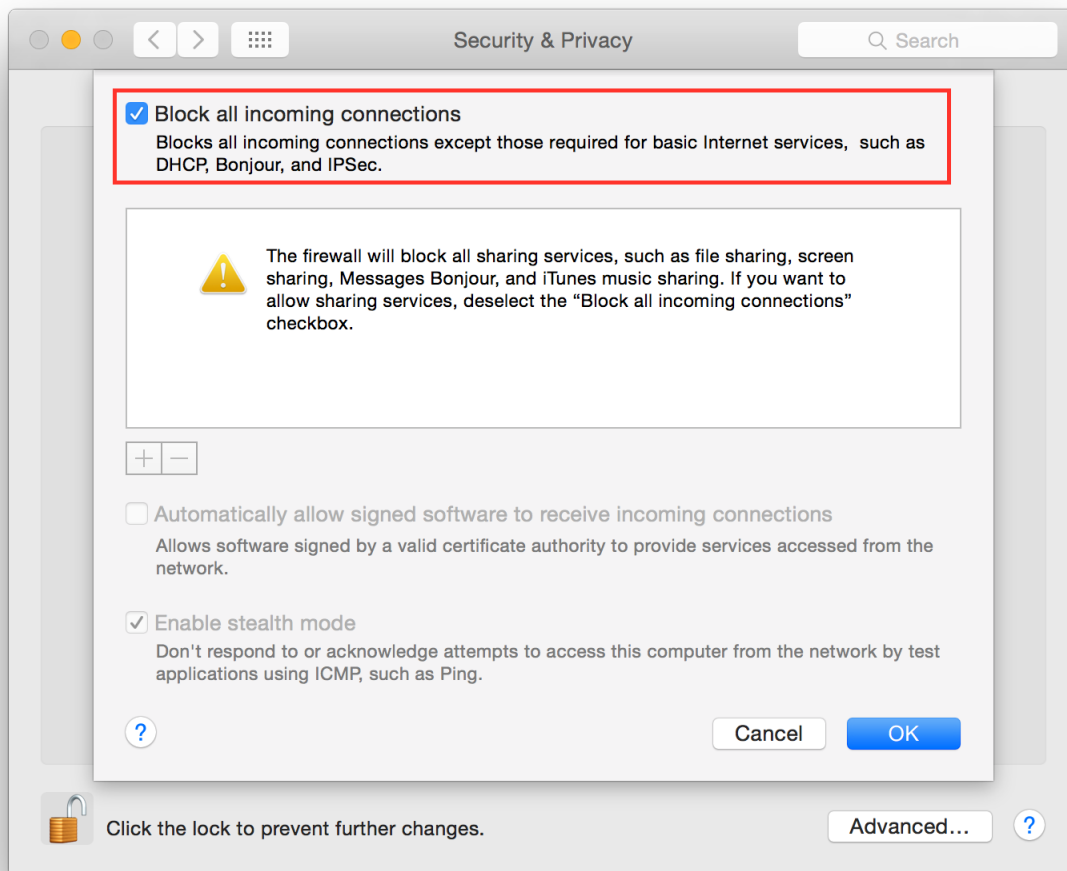
It is suggested to enable the Firewall and have it always running. Go to:

System Preferences Security & Privacy Firewall

Click on “Turn On Firewall”.



Now click on “Firewall options”, a new panel will appear. Click on “Block all incoming connections”.



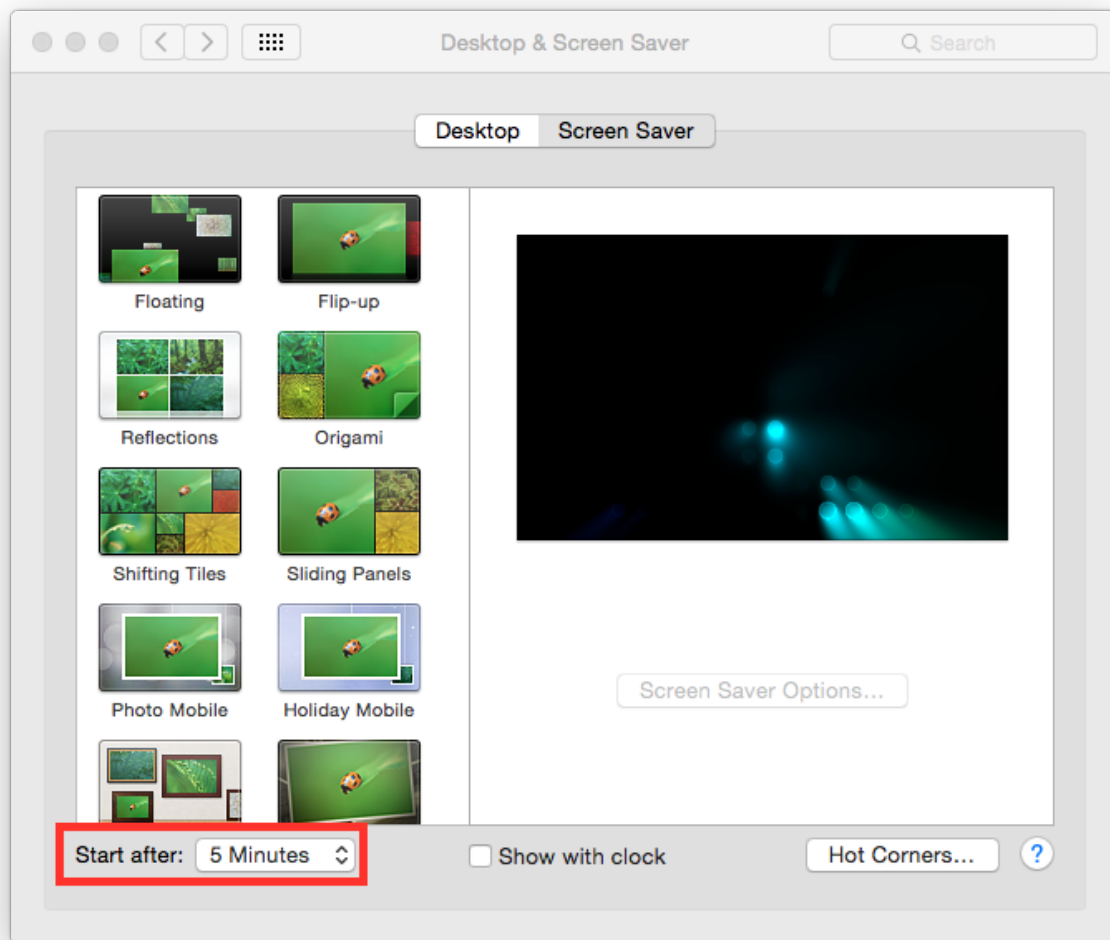
Using “Block all incoming connections” will block all incoming connections to your host. This will block also all sharing services, such as file sharing, screen sharing, Messages Bonjour, iTunes music sharing and other features. If your host is providing any kind of service, this option is not suggested; you should disable it.

### Enable screen saver

It is suggested to enable the screen saver to automatically lock your screen after a while. Go to:

System Preferences Desktop & Screen Saver Screen Saver

Set “Start after” to “5 Minutes”.



## Empty trash securely

When you delete a file, OS X only deletes the index entry for the file, which tells the system the file's contents are free to be overwritten; however, the data still remains and may be recovered using a forensics software. It is a good practice to always empty your trash securely. Your data will be securely wiped from disk in an irreversible way. In the previous OS X releases there was an option to enable safe delete, Apple has removed this feature in OS X El Capitan. However, you can use command line tools.

You can use the `rm` command from Terminal to delete files with the `-P` option, as stated in `man rm` this option is used to:

Overwrite regular files before deleting them. Files are overwritten three times, first with the byte pattern 0xff, then 0x00, and then 0xff again, before they are deleted.

For example if you want to delete `test.pdf` you should open Terminal and use:

```
$ rm -P test.pdf
```

The `srm` command is specifically designed for secure deletion from command line, as stated in `man srm`:

`srm` removes each specified file by overwriting, renaming, and truncating it before unlinking. This prevents other people from undeleting or recovering any information about the file from the command line.

For example if you want to delete *test.pdf* you should open Terminal and use:

```
$ srm test.pdf
```

## Erase free space

In some cases, you might want to run an overwrite task on the free space of a given drive. You can use the *diskutil* command line utility, open Terminal and use:

```
diskutil secureErase freespace LEVEL /Volumes/DRIVE_NAME
```

In this command, change `LEVEL` to a number of 0 through 4, the available options are:

- 0 is a single-pass of zeros
- 1 is a single-pass of random numbers
- 2 is a 7-pass erase
- 3 is a 35-pass erase
- 4 is a 3-pass erase

Change `DRIVE_NAME` to the name of the mount point.

## Power off memory during standby

By default during stand-by memory are kept powered on, this is prone to forensics acquisition of your memory. As stated in *man pmset*:

`hibernatemode` supports values of 0, 3, or 25. Whether or not a hibernation image gets written is also dependent on the values of `standby` and `autopoweroff`

For example, on desktops that support standby a hibernation image will be written after the specified `standbydelay` time. To disable hibernation images completely, ensure `hibernatemode`, `standby` and `autopoweroff` are all set to 0.

`hibernatemode` = 0 by default on desktops. The system will not back memory up to persistent storage. The system must wake from the contents of memory; the system will lose context on power loss. This is, historically, plain old sleep.

`hibernatemode` = 3 by default on portables. The system will store a copy of memory to persistent storage (the disk), and will power memory during sleep. The system will wake from memory, unless a power loss forces it to restore from hibernate image.

`hibernatemode` = 25 is only settable via `pmset`. The system will store a copy of memory to persistent storage (the disk), and will remove power to memory. The system will restore from disk image. If you want “hibernation” - slower sleeps, slower wakes, and better battery life, you should use this setting.

It is suggested to power off memory at stand-by with the following command:

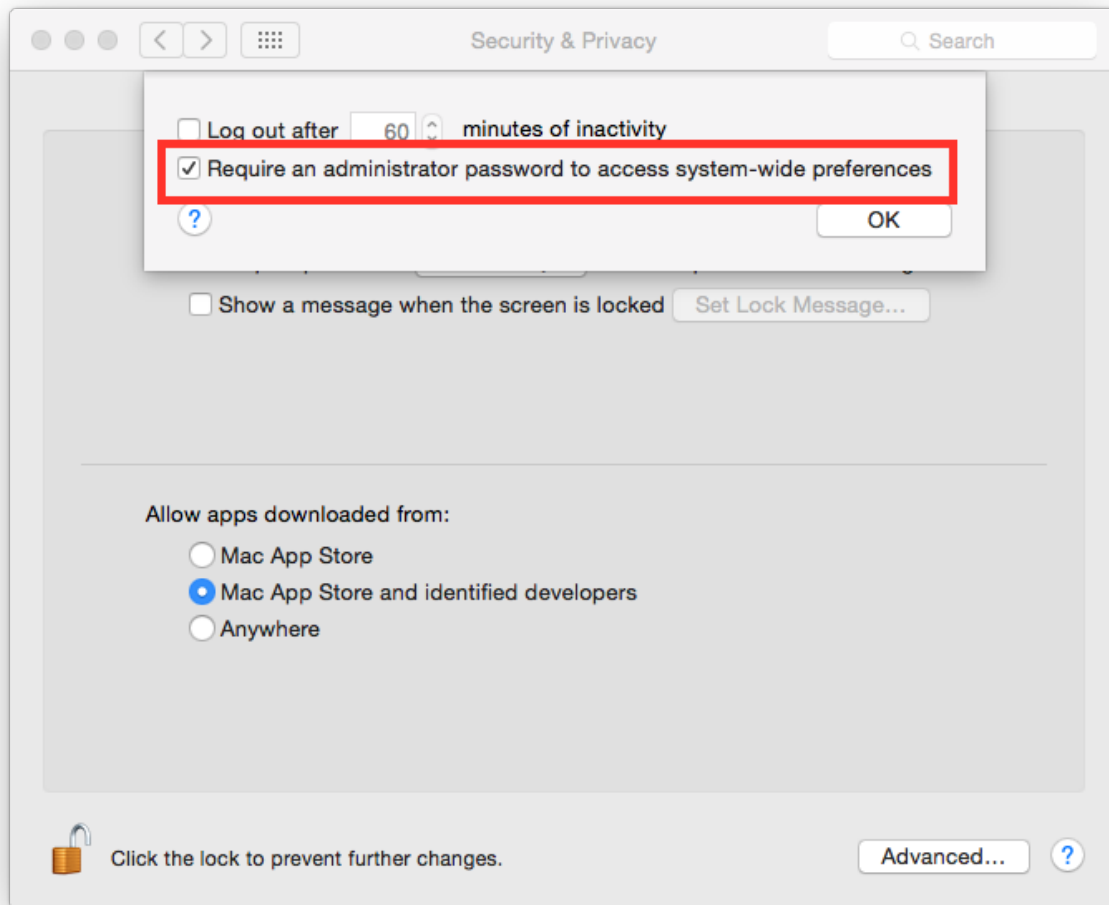
```
sudo pmset hibernatemode 25
```

### Require an administration password

Always require an administration password to access system settings. Go to:

System Preferences Security & Privacy Advanced

Check “Require an administrator password to access system-wide preferences”.

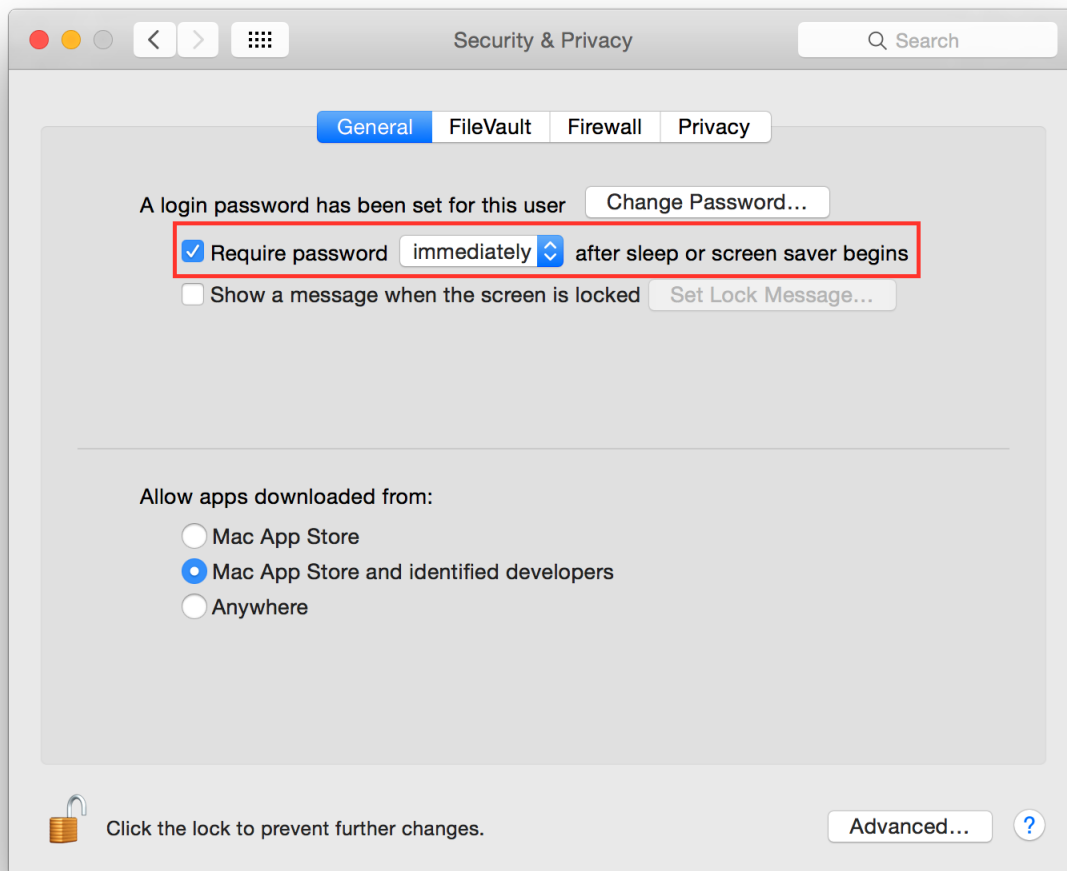


### Require password to un-lock

Requires password to un-lock from sleep or screen saver. Go to:

System Preferences Security & Privacy General

Set “Require password immediately after sleep or screen saver begins”.



## Save to Disk by Default

Many applications bundled in OS X, i.e. Text, save by default new documents to iCloud. It is suggested to set default save target to be a local disk, not iCloud with the following command, open Terminal and type:

```
defaults write NSGlobalDomain NSDocumentSaveNewDocumentsToCloud -bool false
```

## Set a Firmware Password

Enabling an optional firmware password offers an increased level of protection. A firmware password is set on the actual Mac logicboards firmware, it is an EFI password which prevents your Mac from being booted from an external boot volume, single user mode, or target disk mode, and it also prevents resetting of PRAM and the ability to boot into Safe Mode. Years ago firmware passwords could be easily bypassed by removing memory. These days Mac's firmware password isn't easily reset. Apple only suggests to bring your Mac in to an authorized Apple Service Provider and have them do it there.

It is suggested to set a firmware password:

- Power off your Mac and turn it on.

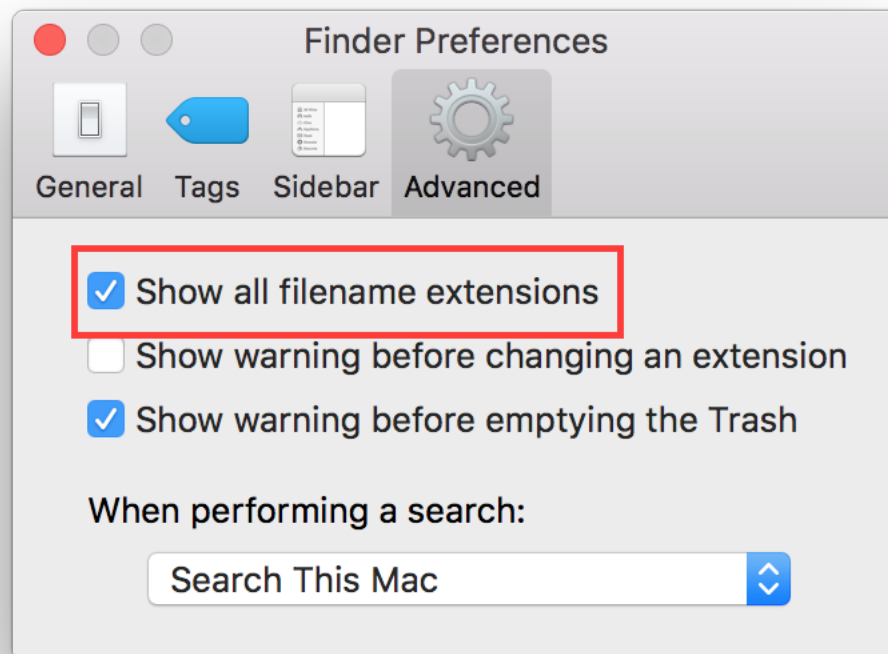
- Activate Recovery Mode (holding down the Command and R keys at boot).
- After a while OS X Utilities will appear.
- Click on the Utilities menu from the menu bar.
- Select Firmware Password Utility.
- Click on ‘Turn On Firmware Password’ and follow the wizard.
- When done, restart your Mac.

### Show all filename extensions

It is a good practice to always show file names extensions. Start Finder app. Go to:

Preferences Advanced

Check “Show all filename extensions”.



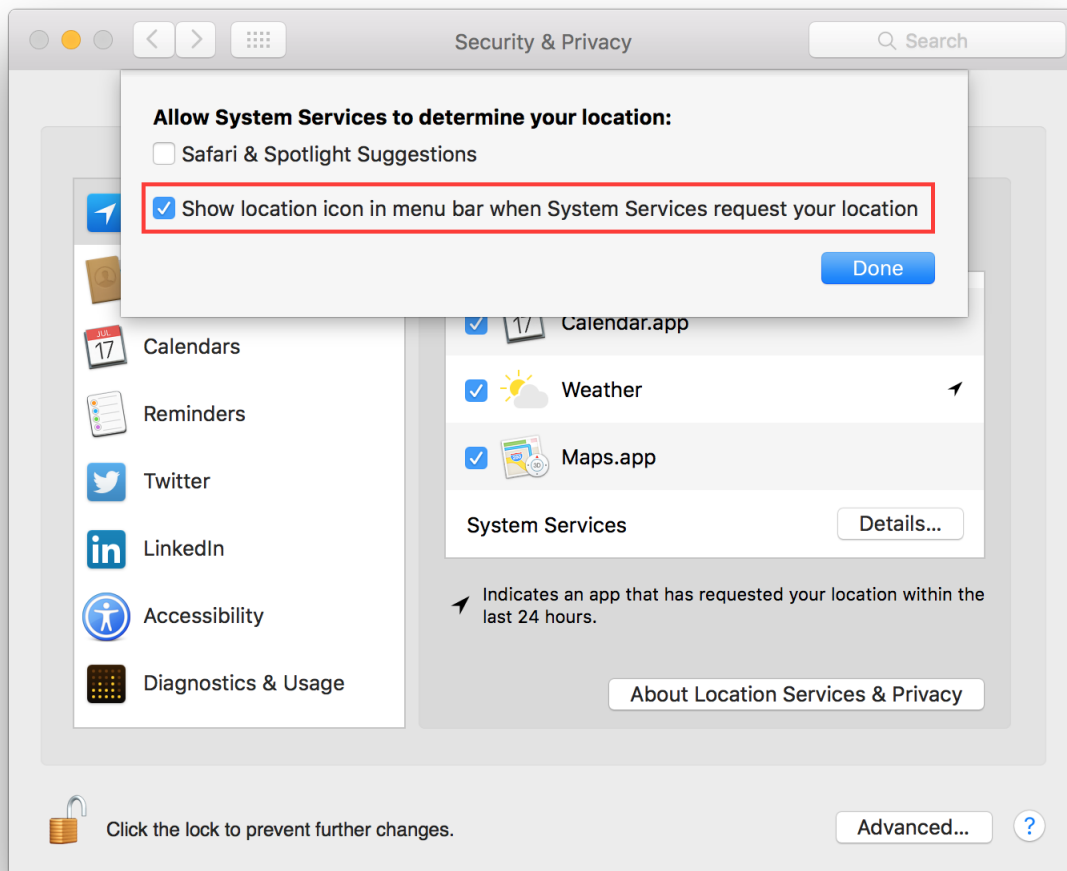
### Show when localization is used

System services could ask to use localization data. It is suggested to show location icon when localization data are requested. Go to:

System Preferences Security & Privacy Privacy Location Services



Select “System Services” and click “Details...”. Check “Show location icon in the menu bar when System Services request your location”.



## Users privilege separation

It is suggested to use different accounts for administration and normal use. Create an account with admin privileges for special tasks and maintenance and a regular user for your normal use. Don't use the same password for both.

## References

- <https://github.com/herrbischoff/awesome-osx-command-line>
- <http://www.frameless.org/2011/09/18/firewire-attacks-against-mac-os-lion-filevault-2-encryption/>



## CHAPTER 2

---

### Contributing

---

Every kind of contribution is really appreciated! Feature requests, suggestions, fixes or documentation contributions are welcome. Please send a patch with your contribution using Github [pull requests](#) or just get in touch with me.

### 2.1 Feedback

Please send questions, comments, suggestions or rants to [alessandro@tanasi.it](mailto:alessandro@tanasi.it) (@jekil).



## CHAPTER 3

---

### License

---

Harden the world project is licensed: [Creative Commons Attribution 4.0 International](#).

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.