Building ClamAV for Mac OS X 10.7 Lion (Server)

Author : Christoph Murauer
Date : 11.11.2011
Version : 1.1
Created using : Apple´s Wiki 3 Server
E - Mail : christoph_murauer@mac.com
Website : http://www.mac.ph

Contents :

1 : Set a firmware password.
2 : Use built-in security.
3 : Your new friend the terminal (only the first contact).
4 : GnuPG 1.4.11.
5 : ClamAV 0.97.3 (including .plist files).

 # 1 : Set a firmware password.

Prerequisites for all following steps :

- A Mac which matches the requirements for Mac OS X 10.7 Lion and a broadband internet connection.
- The latest Mac OS X Lion 10.7.2 version from the Mac App Store.
- Optionally the latest Mac OS X Lion 10.7.2 Server extension from the Mac App Store.
- The latest Xcode 4.2.1 version from the Mac App Store.
- Install (there is no Java and no Flash by default) thees things (without the security risk Flash if possible) and create in your home folder a subfolder called Source.

Security starts at the physical access to your machine (who can use your Mac and boot from which media). A firmware password is also known as EFI password or EFI firmware password. Without this password it is some work to change the default boot device.

Set a EFI password :

1. Switch on your Mac and press the option (alt) key.
2. Choose to boot from the Recovery HD.
3. Choose <Utilities => Firmware Password Utility>.
4. Follow the dialog as shown (also if you like to change a existing EFI password).
5. After a reboot set the default boot media using <Apple menu => Preferences ... => Startup Disc>.

The default boot media can only be changed before booting the system if you hold down the option (alt) key, enter the correct EFI password (at this time the QWERTY keyboard layout is used) and press the return key.

Remove a forgotten EFI password :

1. Switch off your machine and remove the power cord.
2. Change the total amount of RAM in your machine (that means, remove or add some RAM). Don´t try that on a MacBook Air.
3. Switch on your Mac (don´t forget the power cord) and press command (cmd) key + option (alt) key + p + r at the same time and hold it down.
4. Wait till you have heared the chime 3 times and release the keys after the third tone (this deletes the PRAM / NVRAM and the old EFI password).
5. After the password is removed you can restore the old RAM configuration and set a new EFI password.

Attention, a EFI password can´t prevent from a reset, a power off or administrative local / remote access (as shown before, like a sudo nvram or like other tools). So

be careful whom you let play around with your machine.

EFI and the use of the Recovery HD of Mac OS X 10.7 Lion :

1. Switch on your Mac and press the option (alt) key.
2. Enter the EFI password and press the return key.
3. Choose to boot from the Recovery HD.
4. Choose the needed utility (restore the system over internet, repair permissions and ACL´s ...) and follow the instructions as displayed.

 # 2 : Use built-in security.

Let´s activate some built-in security features of the operating system.

<Apple menu => Preferences ... > :

<Security => General>

- Select password required <immediately>.
- Select deactivate automatic login.
- Select demand for a administrator password for preferences panes using a lock symbol.
- I use the lock screen feature instead of the function to log out after x minutes of inactivity. Booth things can be combined.
- Select automatic update the list for save downloads (this is Apple´s malware protection XProtect).
- Deselect deactivate „Reboot in Safari“ if the screen is locked. This is the new guest system which boots with a save system only in a Safari session. There is no Flash available and the guest system requires iCloud and a activated Find My Mac.

<Security => FileFault>

- Leave it deactivated - why ? Because it needs CPU time, disc space, RAM and if someone has your account credentials (username & password) or the master recovery key, then all data is decrypted.
- Use Elettra instead of FileFault to protect your informations (we build it later).

<Security => Firewall>

- Click on <start>.
- Click on <Advanced options ... >.
- Deselect block all incoming connections.
- Deselect allow signed software automatically to accept incomming connections.
- Select Activate Stelth Mode.
- Click on <ok>.

- 2 things as information, first the firewall asks if a application was launched the first time and provide the option to accept incoming network connections (then you can choose to allow or deny it and change it also later under <Advanced options ... > if needed). The second is, that the stealth mode protects against a normal ping but not against LAN / WLAN sniffing and also not against network scans with nmap or Nessus - but it is better then to do nothing.

<Security => Privacy>

- Choose self whether you like to allow applications to use your position (the standard setting is deactivated).
- The new guest system and Find My Mac requires activated location services. If you like to use them then select activate location services.

<Energy Saving => Power Supply>

- Select Wake on demand (also known as wake on lan). If the machine is connected to a power supply, if the Airport card supports it and if you are connectet to a Airport Extreme or TimeCapsule (not all configurations will work) this works also for WLAN.

<iCloud / MobileMe>

- Back to my mac requires the password (there is only 1) of your MobileMe or iCloud account. So be careful what you are doing with it and whom you tell it (don´t forget to configure it also on your Airport Express / Extreme or TimeCapsule). The new iCloud service don´t support shard discs on this devices at the moment.
- The new guest system requires a activated Find My Mac.
- Find My Mac requires activated location services.

<Network => Wi - Fi => Advanced options ... => Wi - Fi (the old name for it was AirPort)>

- Select create computer to computer networks.
- Select change network.
- Select activating and deactivating Wi - Fi.
- Click on <ok> and <apply> to change the settings.
- Check the available options if you use other connection types.

<Bluetooth>

- If possible select invisible.
- If you don´t use / need bluetooth then deactivate it.

<Sharing>

- Run only services which you really need and define users which are allowed to use it if possible.

- Check the options of your running services for example the SSH version and the open ports on all router / firewalls. Check also the configuration files off this applications to hide the used software and it´s version.

<Users & Groups => Password>

- Choose a strong password (containing letters, number, special characters, mix upper and lowercase) which noone knows (also not if the person knows you personally) and don´t write it down.
- Also don´t use the same password as you already use on other (public) services. And change the used passwords from time to time.
- You can use the assistant to get a good password.
- Don´t type in informations to remember the password (password hint).
- Deselect user can change the password using the Apple - ID.

<Users & Groups => Guest Account>

- Deactivate the guest account (this is not the same as the new guest system on the login screen).
- Deselect allow guests to login at this computer.
- Deselect allow a guest to access a shared folder.

<Users & Groups => Login Options>

- Choose what you need / prefer.
- Select show menu for fast user switch and select <symbol>, that needs less space in the menu bar.

<Date & Time => Date & Time>

- Select update date & time automatically and use a trusted NTP server like time.euro.apple.com.

<Software Update => Scheduled Checks>

- Select check for updates and use <daily>.
- Select download updates automatically (the updates are downloaded but not installed, a dialog will appear if something new is there to install).

<TimeMachine>

- Select a media to store your backups. As more backups you have as less data you loose if something bad happened.

<Startup Disc>

- Choose the default boot disc and close the lock.
- This preferences pane can also be used to boot from a FireWire disc using the target disc mode.

<Launchpad => Utilities => Kaychain Access> :

<Keychain Access => Preferences ... => General>

- Select show key ring status in menu bar (a new symbol with a lock appears on the right side of the menubar).
- If you leave your machine select this <lock symbol => lock screen>.

<Launchpad => Safari> :

<Safari => Preferences ... => General>

- Deselect open „secure“ files after download.

Reboot your machine and check whether the settings are set as they should be. Then change all other settings for your needs. The settings for the client and the server version are the same.

 # 3 : Your new friend the terminal.

Many people don´t like or hate the terminal but there is no reason for that. Open <Launchpad => Utilities => Terminal> and use <Terminal => Preferences... > to configure the look and feel to fit your needs. Some details about the terminal.

- The default shell is the bash.
- The terminal is case sensitive. This means that there is a big difference to write man, mAn, Man or whatever.
- One command is one line (even I need two or more lines in the text for one command).
- All entered commands are saved in the file .bash_history in your home folder - after you quit the running terminal session.
- You can view and reuse the entered commands by using the arrow keys up and down.
- There is always a space between the command and its options and arguments.
- Options and / or arguments start with - or --.
- At the end of a command you have to press the return key to get a effect.
- Thats enough for the moments.

The missing Library folder :

One of the new things in Mac OS X 10.7 Lion is, that the Library folder in your home folder is hidden. Open your home folder using <Finder => Go To => Home Folder> to check that. Then go back to your terminal window and enter the following command.

chflags nohidden $HOME/Library

Press the return key and go again back to your home folder in <Finder>. Now the Library folder should be there. Congratulations, your first use of terminal was successful. To get more informations about the terminal use the help menu. To get more informations to a command type in the following in a terminal window.

```
man chflags
```

Use instead of chflags the name of the command for which you need more informations.

4 : The Mac GNU Privacy Guard.

GnuPG is the complete and free implementation of the OpenPGP standard as defined in RFC4880. GnuPG allows to encrypt and sign / decrypt and verify data and communication. I know that there are more tools for GnuPG but at the moment there is no working GPGMail plugin for Lion so GnuPG 1.4.11 is enough for the moment.

Download the source code, check the integrity, extract, build and install :

```
cd $HOME/Source
mkdir gnupg
cd gnupg
curl -L ftp://mirror.switch.ch/mirror/gnupg/gnupg/gnupg-1.4.11.tar.gz -o
gnupg-1.4.11.tar.gz

openssl sha1 gnupg-1.4.11.tar.gz
```

SHA1 checksum : bffb0c60b2e702980f7148ee3a060f29adc82331

```
tar -xzvf gnupg-1.4.11.tar.gz
cd gnupg-1.4.11
./configure
```

For more options use : ./configure --help=short

```
make
make check
sudo make install
```

The command sudo requires always a administrator password like the one from your main account.

gpg was installed in /usr/local/bin.

Import existing keys :

The easiest way is to move the directory .gnupg from the old home folder to the new home folder. Don´t forget to backup .gnupg and don´t loose this directory (because your secret key is also in this directory).

Create a new secret and a new public key (see the man pages to revoke a key) :

gpg --gen-key

And follow the displayed dialog.

Modify the configuration file gpg.conf :

cd $HOME/.gnupg
pico gpg.conf

The following thing must be changed.

Line 151 : keyserver-options auto-key-retrieve

With this option gpg tries to download a key from the key server if possible.

Some useful commands in pico.

- Search : press ctrl + w, enter the search text and press return.
- Cursor position : ctrl + c displays the line number.
- Save : ctrl + o and press return.
- Quit : ctrl + x (if something was modified then pico ask to save or not).

Manage the public (pubring.gpg) and secret (secring.gpg) keys if needed :

gpg --list-secret-keys --fingerprint
gpg --list-public-keys --fingerprint
gpg --delete-key NAME

NAME = are the 8 characters in the first line between the / and the first date. To delete secret keys check man gpg.

Test whether gpg works as it should :

Before we checked the integrity of gpg with a SHA1 checksum. Never ever check gpg with itself. Now after installing we can do that to test gpg.

```
gpg --verify gnupg-1.4.11.tar.gz.sig gnupg-1.4.11.tar.gz
```

This should bring an error which tells you that there is no key on the key server. The reason is that the key which was used for signing is already expired. We download now a .sig file to test it.

```
cd $HOME/Source/gnupg
curl -L ftp://mirror.switch.ch/mirror/gnupg/gnupg/
gnupg-1.4.11.tar.gz.sig -o gnupg-1.4.11.tar.gz.sig
gpg --verify gnupg-1.4.11.tar.gz.sig gnupg-1.4.11.tar.gz
```

If all has worked as it should then there appears a text with the message correct / good signature. From now you can use gpg to check the integrity of downloaded files. For more details see the man pages using man gpg.

 # 5 : ClamAV.

ClamAV is a open source antivirus engine from the company Sourcefire Inc., which can detect trojans, viruses, malware and other bad things (like phishing).

It is a old topic that many people say that a antivirus software is not needed on a Mac. But it is not true - we had malware, trojans (here in Europe the trojan of the state also called „Staatstrojaner“) many other bad software and also phishing.

If you use Mac OS X Lion 10.7 Server there is a built-in ClamAV engine. The version of the engine is outdated but the signatures are up to date. If you run your own mail server the included ClamAV version and the included SpamAssassin version are enough to scan passing mail traffic. So put your fingers away from the built-in engine and don´t touch it. Apple will update that for you, otherwise you have many problems during the next software update.

There is also a nice GUI (graphic user interface) called ClamXAV from Mark Allan but I don´t use it (the developer needs to long to update to new engine versions and to use a own engine work not as it should). So let´s build our own engine and use AppleScript to scan e - mails and monitor folders.

Optionally to update from ClamAV 0.97.2 to ClamAV 0.97.3 :

If you have no old ClamAV engine installed start below this point.
Backup the configuration files from /usr/local/clamav/etc and the signature database from /usr/local/clamav/share/clamav.

Remove the old installation.

cd $HOME/Source/clamav/clamav-0.97.2

The following command works only if you don´t run a make clean or make distclean before - otherwise a rerun of configure (with the original options) is necessary. Created directories arn´t removed.

sudo make uninstall

Remove the old directories.

sudo rm -R /usr/local/clamav
sudo rm -R $HOME/Source/clamav/clamav-0.97.2

The old source code tar balls are still there if you haven´t deleted them before.

Download the source code, check the integrity, extract, build and install :

cd $HOME/Source
mkdir clamav
cd clamav
curl -L http://sourceforge.net/projects/clamav/files/clamav/0.97.3/clamav-0.97.3.tar.gz/download?use_mirror=switch -o clamav-0.97.3.tar.gz

curl -L http://sourceforge.net/projects/clamav/files/clamav/0.97.3/clamav-0.97.3.tar.gz.sig/download?use_mirror=switch -o clamav-0.97.3.tar.gz.sig

gpg --verify clamav-0.97.3.tar.gz.sig clamav-0.97.3.tar.gz
tar -xzvf clamav-0.97.3.tar.gz
cd clamav-0.97.3
./configure --enable-no-cache --enable-dns-fix --prefix=/usr/local/clamav

For more options use : ./configure --help=short
The user and group _clamav must not be specified - configure searchs for it and find it self.

make && make check && sudo make install

In the last part we used for each command one line. Now we use && between the commands. This makes sure that the next step / command is only started if the last was executed successfully.

clamav was installed in /usr/local/clamav.

Modify the configuration file clamd.conf :

sudo pico /usr/local/clamav/etc/clamd.conf

The following things must be changed.

Line 8 : #Example
Line 80 : LocalSocket /tmp/clamd.socket

For the rest I use the ClamAV configuration file of Mac OS X 10.7 Server.

Line 14 : #LogFile /var/log/clamav.log
Line 34 : LogTime yes
Line 43 : LogSyslog yes
Line 48 : LogFacility LOG_LOCAL2
Line 61 : PidFile /var/run/clamd.pid
Line 69 : DatabaseDirectory /usr/local/clamav/share/clamav
Line 88 : LocalSocketMode 660
Line 92 : FixStaleSocket yes
Line 200 : Foreground yes

Syslog requires the following line in /etc/syslog.conf (it is already there in Mac OS X 10.7 Server). If it is not there on the client version then use sudo to edit the file and put it at the end of it.

local2.crit /var/log/clamav.log

Now we try to run clamd and check whether the configuration file works.

sudo /usr/local/clamav/sbin/clamd

This should bring a warning that the virus database is older then 7 days. Check also the log file with <Launchpad => Utilities => Console> and there </var/log => clamav.log>.

Modify the configuration file freshclam.conf :

sudo pico /usr/local/clamav/etc/freshclam.conf

The following things must be changed.

Line 8 : #Example
Line 74 : DatabaseMirror database.clamav.net

For the rest I use the ClamAV configuration file of Mac OS X 10.7 Server.

Line 13 : DatabaseDirectory /usr/local/clamav/share/clamav
Line 17 : #UpdateLogFile /var/log/freshclam.log
Line 29 : LogTime yes
Line 33 : LogVerbose yes
Line 37 : LogSyslog yes
Line 42 : LogFacility LOG_LOCAL2
Line 46 : PidFile /var/clamav/myfreshclam.pid
Line 51 : DatabaseOwner _clamav
Line 99 : Checks 48
Line 138 : Foreground yes
Line 208 : Bytecode no

Optionally to notify clamd after freshclam has updated the database. If clamd is not running then freshclam will report an error (no problem, you can ignore it) that clamd was not successfully notified about the update (see also <Launchpad => Utilities => Console> and there <system.log> or </var/log => freshclam.log>).

Line 121 : NotifyClamd /usr/local/clamav/etc/clamd.conf

Now let´s try to update the database and rerun clamd to check whether the message about the outdated signatures is still there or not (depends whether you have stopped clamd before or not).

sudo /usr/local/clamav/bin/freshclam
sudo /usr/local/clamav/sbin/clamd

If you get the described error then don´t stop clamd with ctrl + c and rerun.

sudo /usr/local/clamav/bin/freshclam

Using .plist files and launchd to start freshclam and clamd at boot time :

To create the files you can use the application you prefer but be sure that you create a plaintext file. So let´s start with the file for clamd.

cd $HOME/Source/clamav
pico net.clamav.clamd.plist

Add the following lines to the file and save it.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
```

```
<plist version="1.0">
<dict>
        <key>Label</key>
        <string>net.clamav.clamd</string>
        <key>KeepAlive</key>
        <true/>
        <key>RunAtLoad</key>
        <true/>
        <key>ProgramArguments</key>
        <array>
                <string>/usr/local/clamav/sbin/clamd</string>
        </array>
        <key>ServiceDescription</key>
        <string>ClamAV 0.97.3 Clamd</string>
</dict>
</plist>
```

Now we create the .plist file for frshclam.

pico net.clamav.freshclam.plist

Add the following lines to this file and save it.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Label</key>
        <string>net.clamav.freshclam</string>
        <key>KeepAlive</key>
        <true/>
        <key>RunAtLoad</key>
        <true/>
        <key>ProgramArguments</key>
        <array>
                <string>/usr/local/clamav/bin/freshclam</string>
                <string>-d</string>
        </array>
        <key>ServiceDescription</key>
        <string>ClamAV 0.97.3 Freshclam</string>
</dict>
</plist>
```

Optionally if you need more options use Apples Xcode and the integrated property list editor (if you use binary .plist files still keep the text files for easy editing).

open net.clamav.clamd.plist
open net.clamav.freshclam.plist

Attention, before we copy the files, we had to change the ownership (for the owner and the group) of it. So if you like to edit the files later use the above commands with sudo.

ls -la
sudo chown root:wheel net.clamav.clamd.plist
sudo chown root:wheel net.clamav.freshclam.plist
ls -la

Copy the files to /Library/LaunchDaemons.

sudo cp ./net.clamav.clamd.plist /Library/LaunchDaemons/ net.clamav.clamd.plist

sudo cp ./net.clamav.freshclam.plist /Library/LaunchDaemons/ net.clamav.freshclam.plist

Reboot your machine. And then we test whether the .plist files are loaded and the applications are running or not.

sudo launchctl list
ps -u root
ps -u _clamav

The -d option for frshclam means that it runs as daemon in the background and checks the database every 30 minutes (see the checks 48 in the feshclam.conf file).

Using .plist files and launchd to watch the home folder of a user and to run a weekly scan over the entire filesystem :

To create the files you can use the application you prefer but be sure that you create a plaintext file. So let´s start to monitor a home folder of a user.

cd $HOME/Source/clamav
pico net.clamav.clamdscanwatchfolder.plist

Add the following lines to the file and save it.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Label</key>
        <string>net.clamav.clamdscanwatchfolder</string>
        <key>KeepAlive</key>
```

```
        <false/>
        <key>RunAtLoad</key>
        <false/>
        <key>UserName</key>
        <string>root</string>
        <key>ProgramArguments</key>
        <array>
                <string>/usr/local/clamav/bin/clamdscan</string>
                <string>/Users/USERNAME</string>
                <string>-l</string>
                <string>/var/log/system.log</string>
        </array>
        <key>WatchPaths</key>
        <array>
                <string>/Users/USERNAME</string>
        </array>
        <key>LowPriorityIO</key>
        <true/>
        <key>ServiceDescription</key>
        <string>Run clamdscan if /Users/USERNAME was modified</
string>
</dict>
</plist>
```

USERNAME = your user name. The bad point is, that launchd can monitor more then one paths but scan only at the given path. So, if you have a big home folder, then create a own .plist file for each location like /Users/USERNAME/Downloads, /Users/USERNAME/Library/Caches and so on. For the scan results I use the system.log file at /var/log/system.log.

Now we create the .plist file for a weekly scan of the entire filesystem.

pico net.clamav.clamdscanweeklyscan.plist

Add the following lines to this file and save it.

```
?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Label</key>
        <string>net.clamav.clamdscanweeklyscan</string>
        <key>KeepAlive</key>
        <false/>
        <key>RunAtLoad</key>
        <false/>
        <key>UserName</key>
        <string>root</string>
        <key>ProgramArguments</key>
```

```
        <array>
                <string>/usr/local/clamav/bin/clamdscan</string>
                <string>/</string>
                <string>-l</string>
                <string>/var/log/system.log</string>
        </array>
        <key>StartCalendarInterval</key>
        <dict>
                <key>Minute</key>
                <integer>30</integer>
                <key>Hour</key>
                <integer>16</integer>
                <key>Weekday</key>
                <integer>5</integer>
        </dict>
        <key>LowPriorityIO</key>
        <true/>
        <key>ServiceDescription</key>
        <string>Run clamdscan every friday at 16 : 30 / 4 : 30 p.
m.</string>
</dict>
</plist>
```

See more details at man launchd.plist. For the scan results I use the system.log file at /var/log/system.log.

Optionally if you need more options use Apples Xcode and the integrated property list editor (if you use binary .plist files still keep the text files for easy editing).

open net.clamav.clamdscanwatchfolder.plist
open net.clamav.clamdscanweeklyscan.plist

Attention, before we copy the files, we had to change the ownership (for the owner and the group) of it. So if you like to edit the files later use the above commands with sudo.

ls -la
sudo chown root:wheel net.clamav.clamdscanwatchfolder.plist
sudo chown root:wheel net.clamav.clamdscanweeklyscan.plist
ls -la

Copy the files to /Library/LaunchDaemons.

sudo cp ./net.clamav.clamdscanwatchfolder.plist /Library/LaunchDaemons/net.clamav.clamdscanwatchfolder.plist

sudo cp ./net.clamav.clamdscanweeklyscan.plist /Library/LaunchDaemons/net.clamav.clamdscanweeklyscan.plist

Reboot your machine. And then we test whether the .plist files are loaded or not.

sudo launchctl list

This jobs are loaded on demand, so will only find informations with ps if they run. You can also use clamdtop to monitor the jobs.

sudo /usr/local/clamav/bin/clamdtop

For more informations about ClamAV see the latest PDF documentation.