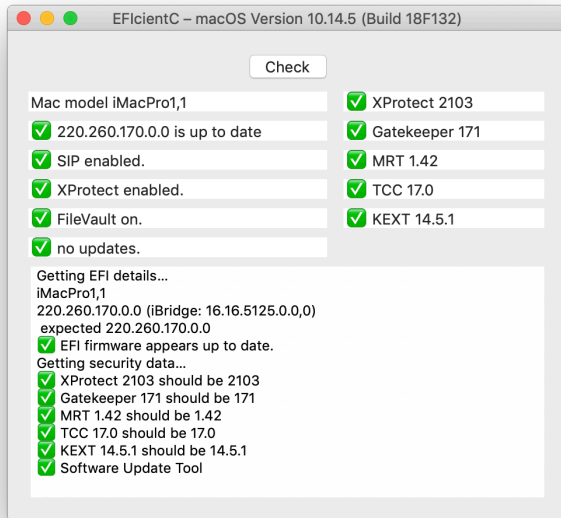


→ [Start](#)

# Start



EFiCienC checks your Mac's key security systems to ensure they're up to date and enabled. This reference explains each item shown in its window.




→ [Check](#)

- [Mac model](#)
- [EFI firmware](#)
- [SIP](#)
- [XProtect](#)
- [FileVault](#)
- [Updates](#)
- [Report](#)
- [XProtect](#)
- [Gatekeeper](#)
- [MRT](#)
- [TCC](#)
- [KEXT](#)
- [Install all updates](#)

# Check

When you first open EFlkienC, it runs its standard checks and completes the result boxes. Click on the **Check** button, or use the **Check** command in the **File** menu, to repeat those checks.

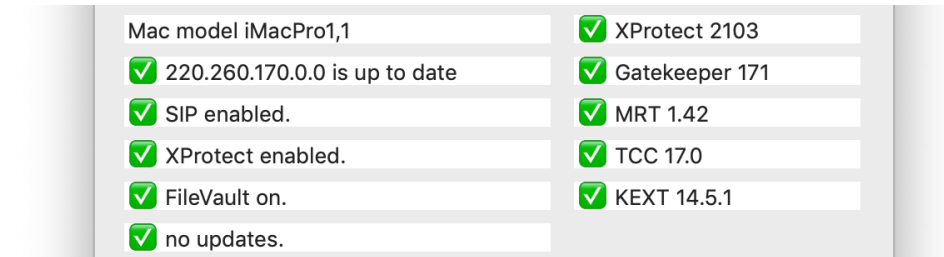
EFlkienC connects to my GitHub server and downloads the current EFI firmware version for this model, then connects to obtain the current list of security data versions. The app checks those versions found on your Mac, compares them, and displays the results.

Those considered to be up to date are prefaced by the emoji  to indicate a 'pass'. Those considered to need further checking or action on your part are prefaced by . Those which appear to be out of date or worth attending to are prefaced by .

The app also connects to Apple's update servers and asks them whether there are security or system updates available for your Mac. This takes longer to complete: while waiting for the result, a circular busy spinner is displayed next to the Updates box. If there are updates available, the **Install all updates** button next to the spinner is shown, so you can decide whether to download and install them.

→ [Install all updates](#)

# Mac model



This displays the specific model of Mac using a standard code, in which *type* of Mac is given first, e.g. MacBookPro, then two digits separated by a comma to identify the *series* and specific *model*, e.g. 10,2. Use these when referring to your Mac so that others can know exactly which it is.

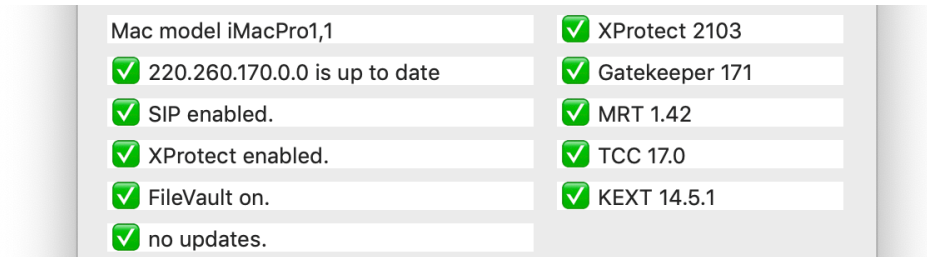
This information is obtained using:

```
let platformExpert = IOServiceGetMatchingService(kIOMasterPortDefault, IOServiceMatching("IOPlatformExpertDevice"))
let modelAsCFString = IORegistryEntryCreateCFProperty(platformExpert, "model" as CFString, kCFAllocatorDefault, 0)
```


It's used to look up the expected EFI firmware version, so if an error occurs when obtaining the Mac model, the EFI firmware version given is almost certainly incorrect too.

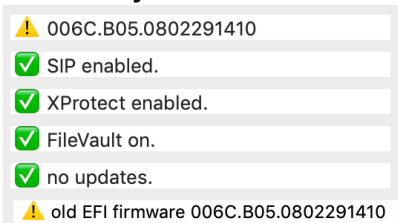
→ [EFI firmware](#)

# EFI firmware



EFIcienc looks up the EFI firmware version currently used by your Mac, and compares it with that believed to be current. If the local version is older than that expected, you will be warned. It's also possible, if you're running a pre-release version of macOS such as a beta, that the local version is newer, in which case no warning is shown.

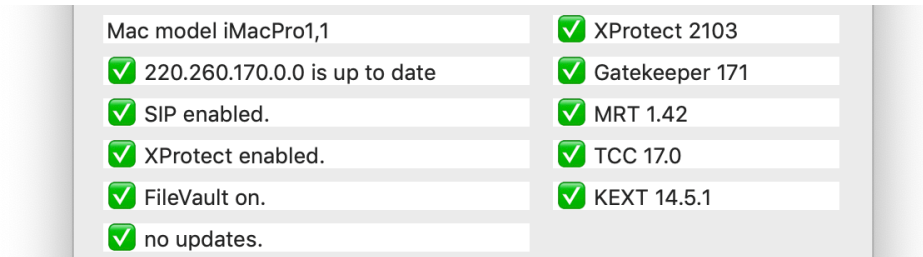
Older systems still use a different version numbering system. EFIcienc can't determine whether those older versions are the latest available, so shows the result with a  warning triangle, so that you can check the version manually:



This information is obtained using

```
let theEntry = IORegistryEntryFromPath(0, "IODeviceTree:/rom")
let efiAsCFString = IORegistryEntryCreateCFProperty(theEntry, "version" as CFString, kCFAllocatorDefault, 0)
```

# SIP



System Integrity Protection or SIP ensures that nothing can tamper with your Mac's system files, and now extends to all the bundled apps in macOS and more besides. Although sometimes it can be helpful to disable SIP, you should never run a Mac for any longer than is essential with SIP turned off.

If SIP is turned off, turn it back on by restarting your Mac in Recovery mode holding Command-R, opening Terminal there and typing in the command

```
csrutil enable; reboot
```

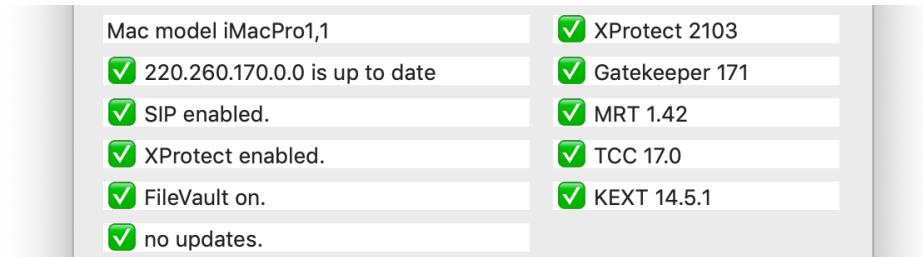
When you press Return, your Mac will then restart in regular mode again with SIP turned back on.

To check SIP, EFlkienC runs the shell command

```
csrutil status
```

→ [XProtect](#)

# XProtect



XProtect is responsible for checking apps and some other files for tell-tale signatures indicating that they are malicious. It should always be enabled: if it's reported in its box at the left to be disabled, contact Apple support as a matter of urgency, as your Mac may have already been attacked by malware. Apple infrequently updates its signature and malware definitions using a pushed security update.

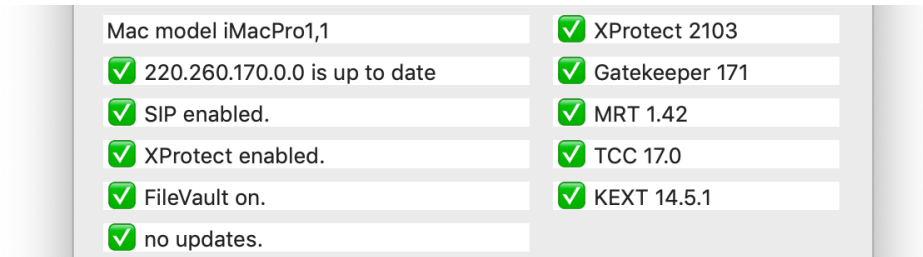
To determine the current version of XProtect data files installed, EFicienC obtains the version number of / System/Library/CoreServices/XProtect.bundle.

To check that XProtect blacklist protection is enabled, it runs the shell command `spctl --status` which should always return that assessments are enabled.

→ [FileVault](#)

→ [Start](#)

# FileVault



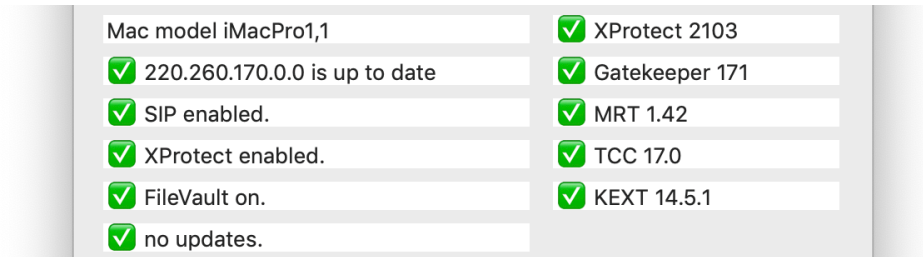
FileVault protects the contents of disks by encrypting them. Internal storage of Macs equipped with T2 chips are always encrypted, although their default encryption doesn't use your password. If there's any risk that someone else could gain access to private or sensitive data on your Mac, you should turn FileVault on. This is an option which you control in the **Security & Privacy** pane of System Preferences.

To check whether FileVault disk encryption is turned on, EFlkienC runs the shell command  
`fdsetup status`

This only applies to the internal storage, although external drives can also be encrypted using FileVault when you wish to protect them.

→ [Updates](#)

# Updates



When EFIenC starts up, and when you click the **Check** button, it connects to Apple's servers and asks them for a list of all system and security updates available for your Mac, using the following command:

```
softwareupdate -l --include-config-data
```

or, in El Capitan,

```
softwareupdate -l
```

This doesn't require you to authenticate, even in El Capitan, and should still work when automatic updates are disabled. When updates are available, this in turn displays the **Install all updates** button, allowing you to download and install them when you wish.

→ [Gatekeeper](#)

→ [Install all updates](#)



# Gatekeeper

Mac model iMacPro1,1	✓ XProtect 2103
✓ 220.260.170.0.0 is up to date	✓ Gatekeeper 171
✓ SIP enabled.	✓ MRT 1.42
✓ XProtect enabled.	✓ TCC 17.0
✓ FileVault on.	✓ KEXT 14.5.1
✓ no updates.	

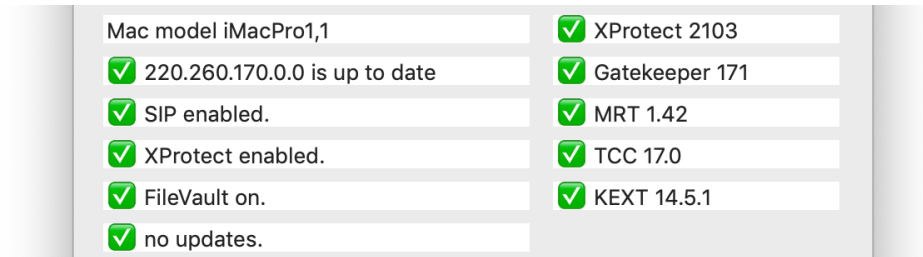
Gatekeeper data files include lists of revoked developer security certificates and other vital information which is used when macOS checks the authenticity of apps and some other items. This is normally performed when the app or item is being run for the first time after being downloaded from the Internet and put into quarantine, but can also be performed on other occasions.

These data files are stored at `/private/var/db/gkopaque.bundle`, and it's that bundle's version number which EFicienC checks and displays. Apple pushes quite frequent updates, every week or two, to ensure that certificate revocations are promulgated promptly.

→ [MRT](#)

→ [Start](#)

# MRT



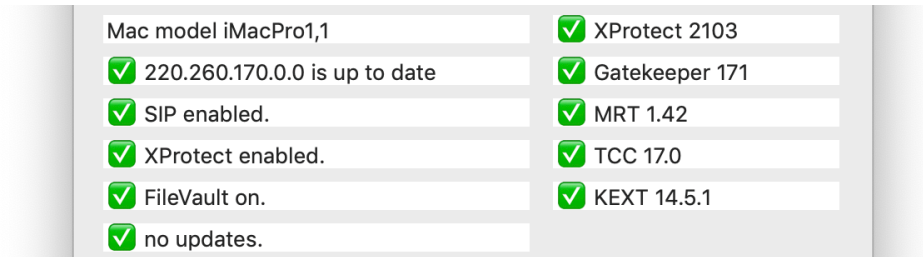
If the system detects that malware is present, it calls on the Malware Removal Tool MRT to do the job. Although this hasn't been updated very often over the last year or so, it remains a central part of macOS system security, and Apple does still maintain it.

The app's data are contained within the app at /System/Library/CoreServices/MRT.app, and the version given here is that of that app.

→ [TCC](#)

→ [Start](#)

# TCC



macOS Mojave introduced new protection for private data in Transparency Consent and Control or TCC. That uses private data which Apple periodically changes using its pushed update service to replace / System/Library/Sandbox/TCC\_Compatibility.bundle.

EFicienC shows the version number of that bundle.

→ [KEXT](#)

→ [Start](#)

# KEXT

Mac model iMacPro1,1	✓ XProtect 2103
✓ 220.260.170.0.0 is up to date	✓ Gatekeeper 171
✓ SIP enabled.	✓ MRT 1.42
✓ XProtect enabled.	✓ TCC 17.0
✓ FileVault on.	✓ KEXT 14.5.1
✓ no updates.	

Prior to macOS 10.15, macOS has used a kernel extension exclude list to prevent some old and conflicting kernel extensions from being loaded. For macOS 10.11 to 10.14, this is obtained from that extension, at / System/Library/Extensions/AppleKextExcludeList.kext.

Catalina uses a different mechanism for blocking kernel extensions, so doesn't show a version here.

→ [Install all updates](#)

→ [Start](#)

# Install all updates

A screenshot of a Mac OS notification bar. On the left, there is a yellow warning triangle icon followed by the text "updates available." in a light gray box. On the right, there is a button with the text "Install all updates" in a rounded rectangle.

The **Install all updates** button only appears when EFlcienC has discovered that there are updates available for your Mac, although you can always force them to be downloaded and installed using the menu command. When you click on this button, the app runs the command:

```
softwareupdate -ia --include-config-data
```

or, in El Capitan,

```
sudo softwareupdate -ia
```

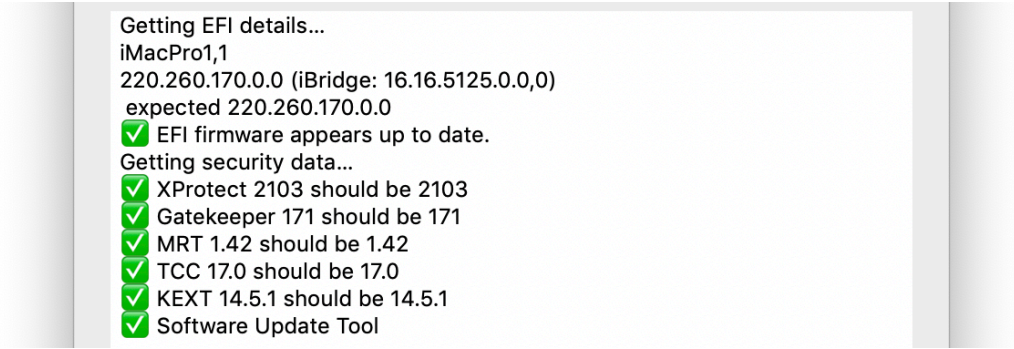
If you are running El Capitan, you need to authenticate before this command can be run, but that is not required in Sierra or later. This tries to connect to Apple's servers, and downloads and installs all pending updates for you.

 This automatically installs *all* pending system and security updates, whether you want them or not.

When large updates are available, it may take several hours to complete, during which EFlcienC will display a 'busy spinner' to indicate that it is still busy. My free app LockRattler allows you to download and install individual updates instead.

→ [Report](#)

# Report



```
Getting EFI details...
iMacPro1,1
220.260.170.0.0 (iBridge: 16.16.5125.0.0,0)
expected 220.260.170.0.0
✅ EFI firmware appears up to date.
Getting security data...
✅ XProtect 2103 should be 2103
✅ Gatekeeper 171 should be 171
✅ MRT 1.42 should be 1.42
✅ TCC 17.0 should be 17.0
✅ KEXT 14.5.1 should be 14.5.1
✅ Software Update Tool
```

In addition to displaying brief information in the boxes above, EFicienC also provides more in the scrolling text area in the lower part of its window. This may include errors encountered when trying to obtain some of those values.

It also lists the latest dates of installation of security data files, which are derived from that Mac's install history at / Library/Receipts/InstallHistory.plist. Those are checked again after the installation of any updates, and should confirm that the update has been correctly received and installed.

⚠️ macOS doesn't return updated version numbers for these security data files until EFicienC has been quit and opened again. To check that an update has been successfully installed, use this installation information rather than the version numbers, which will remain unchanged.

Select all and copy the contents of the report to paste in as plain or rich text, or use the Export... command to save this to a file in plain text.