OpenCore

Reference Manual (0.9.6.7)

[2023.11.08]

8. `EnableWriteUnprotector`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Permit write access to UEFI runtime services code.

   This option bypasses `W^X` permissions in code pages of UEFI runtime services by removing write protection (`WP`) bit from `CR0` register during their execution. This quirk requires `OC_FIRMWARE_RUNTIME` protocol implemented in `OpenRuntime.efi`.

   Note: This quirk may potentially weaken firmware security. Please use `RebuildAppleMemoryMap` if the firmware supports memory attributes table (MAT). Refer to the `OCABC: MAT support is 1/0` log entry to determine whether MAT is supported.

9. `FixupAppleEfiImages`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Fix errors in early Mac OS X boot.efi images.

   Modern secure PE loaders will refuse to load `boot.efi` images from ~~Mac OS X~~ macOS 10.4 ~~and 10.5~~ to 10.12 due to these files containing `W^X` errors (in all versions) and illegal overlapping sections (in 10.4 and 10.5 32-bit versions only).

   This quirk detects these issues and pre-processes such images in memory, so that a modern loader ~~can~~ will accept them.

   Pre-processing in memory is incompatible with secure boot, as the image loaded is not the image on disk, so you cannot sign files which are loaded in this way based on their original disk image contents. Certain firmware will offer to register the hash of new, unknown images - this would still work. On the other hand, it is not particularly realistic to want to start ~~such~~ these early, insecure images with secure boot anyway.

   Note 1: ~~The quirk is only applied to~~ When enabled, this quirk is applied to all Apple-specific ~~'fat' (both~~ Fat binaries (32-bit and 64-bit versions in one image)~~.efi files, and~~ , and to any other Apple-signed boot images that are not being processed for Apple secure boot.

   Note 2: The quirk is never applied during the Apple secure boot path for newer macOS. The Apple secure boot path includes its own separate mitigations for `boot.efi` `W^X` issues.

   Note ~~2~~3: ~~The quirk is only needed for loading Mac OS X~~ This quirk is needed for macOS 10.4 ~~and 10.5, and even then only if~~ to 10.12 (and higher, if Apple secure boot is not enabled), but only when the firmware itself includes a modern, more secure PE COFF image loader. This ~~includes~~ applies to current builds of OpenDuet, and to OVMF if built from audk source code.

10. `ForceBooterSignature`
    Type: `plist boolean`
    Failsafe: `false`
    Description: Set macOS `boot-signature` to OpenCore launcher.

    Booter signature, essentially a SHA-1 hash of the loaded image, is used by Mac EFI to verify the authenticity of the bootloader when waking from hibernation. This option forces macOS to use OpenCore launcher SHA-1 hash as a booter signature to let OpenCore shim hibernation wake on Mac EFI firmware.

    Note: OpenCore launcher path is determined from `LauncherPath` property.

11. `ForceExitBootServices`
    Type: `plist boolean`
    Failsafe: `false`
    Description: Retry `ExitBootServices` with new memory map on failure.

    Try to ensure that the `ExitBootServices` call succeeds. If required, an outdated `MemoryMap` key argument can be used by obtaining the current memory map and retrying the `ExitBootServices` call.

    Note: The need for this quirk is determined by early boot crashes of the firmware. Do not use this option without a full understanding of the implications.