OpenCore

Reference Manual (1.0.~~1~~.2)

[2024.09.29]

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~~ permissions and section errors in macOS `boot.efi` images.

   ~~Modern secure PE loaders will refuse to load `boot.efi` images from~~ Mac OS X ~~10.4 to macOS 10.12 due to these files containing~~ `boot.efi` images contain `W^X` permissions errors (in all versions) and in very old versions additionally contain illegal overlapping sections (~~in~~ affects 10.4 and 10.5 32-bit versions only). Modern secure PE loaders (including the OpenCore loader in current releases of OpenDuet) will refuse to load these images unless additional mitigations are applied.

   This quirk detects these issues and pre-processes such images in memory ~~,~~ so that a modern loader 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 these early, insecure images with secure boot anyway~~If on a system with such a secure loader, this quirk is required to load Mac OS X 10.4 to macOS 10.12, and is required for all newer macOS when `SecureBootModel` is set to `Disabled`.

   Note 1: 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: When enabled, and when not processing for Apple secure boot, this quirk is applied to:

   - All images from Apple Fat binaries (32-bit and 64-bit versions in one image).
   - All Apple-signed images.
   - All images at `\System\Library\CoreServices\boot.efi` within their filesystem.

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

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`