OpenCore

Reference Manual (0.8~~.1~~.2)

[2022.06.07]

Note 2: At this moment `Exclude` is only applied to `prelinkedkernel` and newer mechanisms.

Note 3: In most cases strategy `Exclude` requires the new kext to be injected as a replacement.

7.5 Emulate Properties

1. `Cpuid1Data`
   Type: `plist data`, 16 bytes
   Failsafe: All zero
   Description: Sequence of `EAX`, `EBX`, `ECX`, `EDX` values to replace `CPUID (1)` call in XNU kernel.

   This property primarily meets three requirements:

   - Enabling support for an unsupported CPU model (e.g. Intel Pentium).
   - Enabling support for a CPU model not yet supported by a specific version of macOS (typically old versions).
   - Enabling XCPM support for an unsupported CPU variant.

   Note 1: It may also be the case that the CPU model is supported but there is no power management supported (e.g. virtual machines). In this case, `MinKernel` and `MaxKernel` can be set to restrict CPU virtualisation and dummy power management patches to the particular macOS kernel version.

   Note 2: Only the value of `EAX`, which represents the full CPUID, typically needs to be accounted for and remaining bytes should be left as zeroes. The byte order is Little Endian. For example, `C3 06 03 00` stands for CPUID `0x0306C3` (Haswell).

   Note 3: For XCPM support it is recommended to use the following combinations. Be warned that one is required to set the correct frequency vectors matching the installed CPU.

   - Haswell-E (`0x0306F2`) to Haswell (`0x0306C3`):
     `Cpuid1Data`: `C3 06 03 00 00 00 00 00 00 00 00 00 00 00 00 00`
     `Cpuid1Mask`: `FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00`
   - Broadwell-E (`0x0406F1`) to Broadwell (`0x0306D4`):
     `Cpuid1Data`: `D4 06 03 00 00 00 00 00 00 00 00 00 00 00 00 00`
     `Cpuid1Mask`: `FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00`
   - Comet Lake U62 (`0x0A0660`) to Comet Lake U42 (`0x0806EC`):
     `Cpuid1Data`: `EC 06 08 00 00 00 00 00 00 00 00 00 00 00 00 00`
     `Cpuid1Mask`: `FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00`
   - Rocket Lake (`0x0A0670`) to Comet Lake (`0x0A0655`):
     `Cpuid1Data`: `55 06 0A 00 00 00 00 00 00 00 00 00 00 00 00 00`
     `Cpuid1Mask`: `FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00`
   - Alder Lake (`0x090672`) to Comet Lake (`0x0A0655`):
     `Cpuid1Data`: `55 06 0A 00 00 00 00 00 00 00 00 00 00 00 00 00`
     `Cpuid1Mask`: `FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00`

   Note 4: Be aware that the following configurations are unsupported by XCPM (at least out of the box):

   - Consumer Ivy Bridge (`0x0306A9`) as Apple disabled XCPM for Ivy Bridge and recommends legacy power management for these CPUs. `_xcpm_bootstrap` should manually be patched to enforce XCPM on these CPUs instead of this option.
   - Low-end CPUs (e.g. Haswell+ Pentium) as they are not supported properly by macOS. Legacy workarounds for older models can be found in the `Special NOTES` section of acidanthera/bugtracker#365.

2. `Cpuid1Mask`
   Type: `plist data`, 16 bytes
   Failsafe: All zero
   Description: Bit mask of active bits in `Cpuid1Data`.

   When each `Cpuid1Mask` bit is set to 0, the original CPU bit is used, otherwise set bits take the value of `Cpuid1Data`.

3. `DummyPowerManagement`
   Type: `plist boolean`
   Failsafe: `false`
   Requirement: 10.4-12
   Description: Disables `AppleIntelCpuPowerManagement`.

Failsafe: Empty
Description: Patches data on specified macOS version or older.

Note: Refer to the `Add MaxKernel` description for matching logic.

11. `MinKernel`
Type: `plist string`
Failsafe: Empty
Description: Patches data on specified macOS version or newer.

Note: Refer to the `Add MaxKernel` description for matching logic.

12. `Replace`
Type: `plist data`
Failsafe: Empty
Description: Replacement data of one or more bytes.

13. `ReplaceMask`
Type: `plist data`
Failsafe: Empty (Ignored)
Description: Data bitwise mask used during replacement. Allows fuzzy replacement by updating masked (set to non-zero) bits. Must be equal to `Replace` in size if set.

14. `Skip`
Type: `plist integer`
Failsafe: `0` (Do not skip any occurrences)
Description: Number of found occurrences to skip before replacements are applied.

7.8 Quirks Properties

1. `AppleCpuPmCfgLock`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.4-12
Description: Disables `PKG_CST_CONFIG_CONTROL` (`0xE2`) MSR modification in AppleIntelCPUPowerManagement.kext, commonly causing early kernel panic, when it is locked from writing.

Some types of firmware lock the `PKG_CST_CONFIG_CONTROL` MSR register and the bundled `ControlMsrE2` tool can be used to check its state. Note that some types of firmware only have this register locked on some cores. As modern firmware provide a `CFG Lock` setting that allows configuring the `PKG_CST_CONFIG_CONTROL` MSR register lock, this option should be avoided whenever possible.

On APTIO firmware that do not provide a `CFG Lock` setting in the GUI, it is possible to access the option directly:

(a) Download UEFITool and IFR-Extractor.
(b) Open the firmware image in UEFITool and find `CFG Lock` unicode string. If it is not present, the firmware may not have this option and the process should therefore be discontinued.
(c) Extract the `Setup.bin` PE32 Image Section (the UEFITool found) through the `Extract Body` menu option.
(d) Run IFR-Extractor on the extracted file (e.g. `./ifrextract Setup.bin Setup.txt`).
(e) Find `CFG Lock, VarStoreInfo (VarOffset/VarName):` in `Setup.txt` and remember the offset right after it (e.g. `0x123`).
(f) Download and run Modified GRUB Shell compiled by brainsucker or use a newer version by datasone.
(g) Enter `setup_var 0x123 0x00` command, where `0x123` should be replaced by the actual offset, and reboot.

Warning: Variable offsets are unique not only to each motherboard but even to its firmware version. Never ever try to use an offset without checking.

On selected platforms, the `ControlMsrE2` tool can also change such hidden options. Pass desired argument: `lock`, `unlock` for `CFG Lock`. Or pass `interactive` to find and modify other hidden options.

As a last resort, consider patching the BIOS (for advanced users only).