Lilu - Generic kext patcher (neue Grundlage für diverse Kexte)

Beitrag von „al6042“ vom 22. Februar 2017, 21:05

Die Bastler von AppleALC und Shiki haben aus den einzelnen Kexten den sogenannten Lilu.kext, wohl den eigentlichen Kext-, Prozess- und Library-Patcher, herausgelöst und als eigenen Kext bereitgestellt.

Ohne diesen, lassen sich AppleALC, ab Version 1.1.0, oder Shiki, ab Version 2.0.0, nicht mehr aus der EFI heraus laden.
EDIT:
Hier findet ihr eine von @NoirOSX zusammengefasste Liste aller Kexte, die Lilu benötigen -> Lilu & Plugins mit Bootflags und Beispielen

Ich bin mal gespannt, wielange diese Art der Abhängigkeit bestand haben wird... aber aktuell geht es nicht ohne.

Beitrag von „al6042“ vom 2. April 2017, 10:54

Und auch hier gehts weiter...

Folgende Neuerung wurden eingebaut:
- Added support for patching different sections/segments
- Added file writing API by lvs1974
- Added strrchr API
- Changed requestAccess to include API version to workaround kext loading issues
- Updated capstone to 3.0.5 rc2
- Improved 32-bit userspace patcher
- Enforced -liluslow in installer and recovery

Ohne diesen, lassen sich AppleALC, ab Version 1.1.x, oder Shiki, ab Version 2.0.x, nicht mehr aus der EFI heraus laden.

Beitrag von „al6042“ vom 13. Mai 2017, 16:53

Ganz frisch aufgetischt...
Version 1.1.1 mit folgenden Neuerungen:
- Changed loading policy to ignore kexts that are not permitted to load
- Increased executable memory buffer from 256 to 1024 bytes
- Allowed different plugins load the same kexts

Beitrag von „Noir0SX“ vom 2. Juni 2017, 20:44

vor ein paar Stunden
v1.1.3

- Reduced binary size by modding capstone
- Fixed LiluAPI::onProcLoad return code
- Added MachInfo::setRunningAddresses for userspace symbol solving
- Added getKernelMinorVersion for symmetry
- Added kernel write protection and interrupt state validation

Note: Please ensure all your plugins contain static OSBundleCompatibleVersion Info.plist property. See details at LiluFriend.

Beitrag von „Noir0SX“ vom 6. Juni 2017, 17:58

Was Neues ...

v1.1.4

- Slightly improved userspace patcher speed for 10.12
- Added missing dyld_shared_cache detection with a fallback
- Defined High Sierra kernel version

Beitrag von „Noir0SX“ vom 10. Juni 2017, 00:05

Configuration

- Add -liludbg to enable debug printing (available in DEBUG binaries).
- Add -liluoff to disable Lilu.
- Add -liluslow to enable legacy user patcher.
- Add -lilulowmem to disable kernel unpack (disables Lilu in recovery mode).
- Add -lilubeta to enable Lilu on unsupported os versions.

Beitrag von „Noir0SX“ vom 29. Juni 2017, 23:52

v1.1.5

- Increased executable memory buffer to page size
- Added auth-root-dmg support High Sierra installer detection (thx Piker-Alpha)
- Added -liluforce to force enable Lilu in safe mode and recovery
- Added preliminary Xcode 9 compatibility

- Added support for unloadable kexts
- Merged official capstone patches up to c508224

Beitrag von „Noir0SX“ vom 11. August 2017, 20:56

Schon paar Tage auf dem Markt, nur hier hat es keiner aktuallisiert

v1.1.6

- Ignored disabled kexts earlier for speed reasons
- Added High Sierra to the list of compatible OS
- Added arrsize API
- Made patch count warning only show in debug mode
- Made kinfo not found logging only show in debug mode
- Added routeBlock API for opcode-based routing
- Centralised user and kernel patcher start time
- Added c-compliant kern_os_cfree implementation
- Added a workaround for page fault kernel panics
- Added a workaround for xnu printf limitations

Beitrag von „Noir0SX“ vom 27. August 2017, 20:52

v1.1.7

- Merged advanced disassembly API (thx Pb and others)
- Added HDE disassembler for quick instruction decoding (by Vyacheslav Patkov)
- Updated capstone to 3.0.5 rc3
- Fixed load API lock type preventing dynamic memory allocation (thx Pb)
- Added setInterrupts API
- Added an option to define custom plugin entry points
- Added const reference evector API
- Added FAT_CIGAM Mach-O support
- Added WIOKit::getComputerInfo API and improved some other WIOKit APIs
- Added support of storing larger than pointer types in evector
- Added -lilubetaall boot argument to skip version checking for all plugins

Beitrag von „al6042“ vom 20. Oktober 2017, 20:34

Auch hier noch die Version 1.2.0 mit folgenden Neuerungen:

Achtung: Nicht alle aktuelle Plugin-Kexte sind mit Lilu 1.2.0 kompatibel. Eine Liste der funktionierenden Typen findet ihr im folgenden Link -> https://github.com/vit9696/Lilu/blob/master/KnownPlugins.md

- Added more handy reporting macros
- Enabled Lilu in safe mode by default with all plugins required to declare supported environments
- Added lzss compression API
- Added crypto and nvram API
- Added support for solving kext symbols from kextcache
- Added memfunc wrappers (e.g. lilu_os_memcpy) to avoid undefined builtins from 10.13 SDK
- Added -liludbgall boot argument (to be on par with -lilubetaall)
- Added unexact process path matching
- Changed compression API logic to support preallocated buffers
- Changed memory allocation logic in certain APIs
- Changed kernel protection API to accept a lock for cpu preemption control
- Changed KextInfo structure to handle disabled and fsonly kexts
- Changed logging API to enforce more proper style
- Disabled advanced disassembly APIs by default (create an issue if you need them)
- Fixed a memory issue in WIOKit::getComputerInfo introduced in 1.1.7
- Fixed several assertions triggering in 10.13 development kernel
- Fixed Xcode 9 compiled binary compatibility with older OS
- Fixed FAT_CIGAM and FAT_MAGIC parsing issues
- Fixed a number of potential memory issues in mach parsing code
- Fixed debug and development kextcache loading issues
- Fixed shutdown issues in -lilulowmem mode
- Fixed seldom boot slowdown when disabling the plugins via boot arguments

Beitrag von „anonymous_writer“ vom 7. November 2017, 22:30

Hier die Version v1.2.1

v1.2.1
Fixed a rare kernel panic when running Lilu with -liludbg
Added a workaround for 10.13.2 beta issues
Fixed compilation with Xcode 8.2
Added prelink usage detection to avoid confusing different kernels
Disabled prelink usage for kext address solving by default since it caused many issues

---

Beitrag von „anonymous_writer“ vom 25. November 2017, 09:28

Hier Vorab schon mal die nächste noch nicht veröffentliche Version.
Source Code vom 25.11.2017

v1.2.2
Acknowledged macOS Install Data and com.apple.recovery.boot prelinkedkernel paths (thx Piker-Alpha)
Fixed ignoring kcsuffix=<suffix> for kexts and less common names

---

Beitrag von „Noir0SX“ vom 25. November 2017, 09:43

> Zitat von G4_Hacker
>
> Source Code vom 25.11.3017

Wird da auch nicht mehr so schnell was kommen, so alt werden wir alle nicht. 

Beitrag von „anonymous_writer“ vom 25. November 2017, 09:45

Sorry Bin der Zeit voraus. Habe es geändert.

Beitrag von „Noir0SX“ vom 7. April 2018, 19:30

v1.2.3

- Added CPU information API for cpu families and generations
- Added IGPU information API for framebuffers and stuff
- Added WIOKit::renameDevice API for device renaming with compatible fixing
- Added KernelPatcher::routeVirtual API for virtual function swapping
- Added PCI register and address manipulation API
- Added basic process modification API
- Added plugin IOService access
- Added address-printing macros
- Added address validation API
- Added strict kext UUID validation to workaround broken kextcache
- Added version info reporting to IORegistry for Lilu and plugins
- Fixed several inline function definitions
- Fixed crash when loading user patches with no binary patches
- Reduced long patch length in function routing API

Beitrag von „Noir0SX“ vom 1. Juli 2018, 19:40

v1.2.4

- Internalize new APIs from 1.2.3
- Added new EFI runtime API with custom variable extensions
- Added new RTC storage API
- Added centralised entitlement hooking API
- Added lilu_os_qsort export (the supported interface is Apple-private)
- Added liludelay=1000 boot argument to insert a 1s delay after each print
- Added new symbol routing API with simplified interface
- Fixed a kernel panic in userspace patching code on 10.14b1

Beitrag von „Noir0SX“ vom 20. Juli 2018, 10:38

v1.2.5

- Added new DeviceInfo API
- Added checkKernelArgument API
- Added enforced LiluAPI interfaces
- Added KextInfo::switchOff API
- Added cpuid API
- Allowed for onKextLoad to accept no callback
- Removed GPU detection code from CPUInfo API
- Enabled by default on 10.14

https://github.com/acidanthera/Lilu/

Beitrag von „anonymous_writer“ vom 26. Juli 2018, 08:49

v1.2.6

> Added Cannon Lake and Ice Lake definitions
> Added missing typed getOSData APIs
> Added -liluuseroff boot-arg to disable user patcher (for e.g. shared cache manipulation)
> Added lilucpu=N boot-arg to assume CPU generation
> Added CPU topology detection APIs>
> Fixed routeMultiple kernel panic and log report
> Switched to Apple lzvn implementation

Beitrag von „Noir0SX“ vom 11. September 2018, 13:28

v1.2.7

- Added support for detecting optimus switch-off
- Added Sanitize target with ubsan support (thx to NetBSD)
- Added disk log dump in DEBUG builds via liludump=N boot-arg (requires plugin rebuild)
- Fixed multiple Mach-O parsing issues

- Fixed support of PCI devices without compatible property
- Fixed PCI class-code masking not detecting HDEF devices

Beitrag von „Noir0SX“ vom 30. Oktober 2018, 16:18

v1.2.8

- Fixed CPU generation detection for Coffee Lake-U
- Fixed PEGP detection with 3D Controller `class-code`
- Fixed userspace patcher compatibility with macOS Mojave
- Allow manually specified reservation in `evector`
- Improved version information printing in DEBUG builds

Beitrag von „Noir0SX“ vom 2. Januar 2019, 15:00

v1.3.1

- Lowered version compatibility to 1.2.0 to let plugins load

Note: This release is functionally not different from 1.3.0, but it fixes plugin loading from `/Library/Extensions` if absolutely necessary.

Beitrag von „Noir0SX“ vom 19. Februar 2019, 12:23

v1.3.4

- Added implicit `eraseCoverageInstPrefix` to `routeMultiple`
- Fixed user patcher kernel panic when running process via `posix_spawn` without exec
- Fixed user patcher codesign issues on recent 10.14 versions with SIP
- Changed `kern_start` and `kern_stop` to contain product prefix to avoid collisions

v1.3.3

- Added support for modern AMD device scanning by **@AlGreyy**
- Added support for VMware device scanning
- Extended supported firmware vendor list

Beitrag von „Noir0SX“ vom 22. März 2019, 14:28

v1.3.5

- Fixed analog audio device detection on certain laptops with Insyde firmware

---

Beitrag von „Noir0SX“ vom 24. Mai 2019, 17:23

v1.3.6

- Lilu now uses OpenCore NVRAM variable GUIDs
- Add support for `0x3E980003` frame id for CFL refresh

Beitrag von „Noir0SX“ vom 3. Juli 2019, 15:45

v1.3.7

- Allow loading on 10.15 without `-lilubetaall`
- Add support for Xcode 11 analysis tools
- Add workaround to 10.15 SDK Dispatch method (use old Xcode when possible)

Beitrag von „Noir0SX“ vom 11. August 2019, 14:50

v1.3.8

- Compile Xcode 11 OSObject stubs into plugins to allow mixing compilers
- Unified release archive names
- Added multirouting support to routeFunction API enabling functions to have multiple proxies
- Added explicit routing type to routeFunction APIs
- Made Lilu use long function routes to ease third-part multirouting

---

Beitrag von „Noir0SX“ vom 31. Oktober 2019, 12:29

v1.3.9

- Added QEMU/KVM vendor compatibility to device detection logic

---

Beitrag von „Noir0SX“ vom 3. Dezember 2019, 05:25

v1.4.0

- Fixed mishandling user patches process list after processKernel API call
- Fixed extra I/O in user patcher even when no patches were needed
- Added support for per-process (LocalOnly) userspace patches

---

Beitrag von „Noir0SX“ vom 13. Januar 2020, 17:22

v1.4.1

- Made applyLookupPatch support kernel patches by passsing null kext
- Export hde64 interface
- Added evector deleter without copying for improved performance
- Allow C strings as module prefix argument to the logging functions

---

Beitrag von „Noir0SX“ vom 2. März 2020, 18:15

v1.4.2

- Fixed IMEI device detection on some platforms
- Added CometLake CPU model support (thx **@stormbirds**)
- Added getFatOffset MachO API

Beitrag von „Noir0SX“ vom 6. April 2020, 15:22

v1.4.3

- Improved modern CPUID detection
- Added BaseDeviceInfo API with improved performance
- Deprecated CPUInfo::getGeneration, WIOKit::getComputerModel(), WIOKit::getComputerInfo()

Beitrag von „Noir0SX“ vom 4. Mai 2020, 15:45

v1.4.4

- Added new CFL connector-less framebuffers: 0x9BC80003, 0x9BC50003, 0x9BC40003
- Fixed KDK support disrespecting file suffixes

Beitrag von „Noir0SX“ vom 1. Juni 2020, 16:30

v1.4.5

- Fixed newer CPU generation detection
- Added failsafe versions of CML framebuffers

Beitrag von „Noir0SX“ vom 3. August 2020, 17:00

v1.4.6

- Added preliminary definitions for 11.0 support
- Temporarily disabled user patcher for 11.0
- Added `external-audio` property to ignore PCI audio cards
- Added in-memory symbol solving for 11.0
- Fixed accidentally solving stabs instead of normal symbols
- Added device publishing API to monitor device startup
- Added DeviceInfo caching for improved performance
- Added implicit slotted (medium) patches in KC mode to reduce patch size

---

Beitrag von „Raptortosh“ vom 9. April 2021, 10:10

v1.5.2

- Fixed AZAL recognition as GPU audio on certain AMD platforms (thx to wkpark)
- Added external GPU disabling API with device and kernel selection via properties
- Added identifiers for Rocket Lake and Tiger Lake CPUs
- Added API to disable builtin GPU (IGPU)
- Reduced hardware presence bruteforce to a more sensible value