Kemon: An Open-Source Pre and Post Callback-Based Framework for macOS Kernel Monitoring

wang yu

Black Hat US'18 Arsenal

- About me

- Background

The Existing macOS Kernel Monitoring Infrastructures

Kernel Authorization Subsystem

https://developer.apple.com/library/archive/technotes/tn2127/_index.html

1. These callback interfaces lack the necessary maintenance and have not been upgraded for about thirteen years.

2. For KAUTH_SCOPE_FILEOP listeners, there are only seven file operation related callbacks available which is obviously not enough.

3. For KAUTH_SCOPE_FILEOP listeners, they are unable to block any file operations.

Kernel Authorization Subsystem (cont)

https://developer.apple.com/library/archive/technotes/tn2127/_index.html

4. For some specific callbacks, input parameters often lack critical context information.

For example, for process creation callback handler, the input parameter is missing command line information.

5. For KAUTH_SCOPE_VNODE listeners, not every file system operation triggers an authorization request.

For example, if an actor successfully requests KAUTH_VNODE_SEARCH on a directory, the system may cache that result and grant future requests without invoking listeners for each one.

Mandatory Access Control Policy

https://developer.apple.com/library/archive/qa/qa1574/_index.html

Q: Why isn't the kernel's MAC framework documented?

A: The kernel's MAC (Mandatory Access Control) framework is not supported for third party development on current systems. The headers were mistakenly included in the Kernel framework installed by the Mac OS X 10.5 SDK (r.5645458).

Mandatory Access Control Policy (count)

CASE 1. Interfaces were deleted or replaced directly

G:\mac_policy\mac_policy_v32.h

```
    mpo_vnode_notify_rename_t       *mpo_vnode_notify_rename;

    mpo_thread_label_init_t         *mpo_thread_label_init;
    mpo_thread_label_destroy_t      *mpo_thread_label_destroy;

    mpo_system_check_kas_info_t     *mpo_system_check_kas_info;
```

G:\mac_policy\mac_policy_v37.h

```
    mpo_vnode_notify_rename_t       *mpo_vnode_notify_rename;

    mpo_reserved_hook_t             *mpo_reserved32;
    mpo_reserved_hook_t             *mpo_reserved33;

    mpo_system_check_kas_info_t     *mpo_system_check_kas_info;
```

G:\mac_policy\mac_policy_v47.h

```
    mpo_system_check_kas_info_t     *mpo_system_check_kas_info;

    mpo_proc_check_cpumon_t         *mpo_proc_check_cpumon;
```

G:\mac_policy\mac_policy_v52.h

```
    mpo_system_check_kas_info_t     *mpo_system_check_kas_info;

    mpo_vnode_check_lookup_preflight_t  *mpo_vnode_check_lookup_preflight;
```

Mandatory Access Control Policy (count)

CASE 2. Prototypes and input parameters were changed directly

G:\mac_policy\mac_policy_v47.h

```
/**
  @brief Access control check after determining the code direc
  @param vp vnode vnode to combine into proc
  @param label label associated with the vnode
  @param cs_blob the code signature to check
  @param cs_flags update code signing flags if needed

  @param flags operational flag to mpo_vnode_check_signature
  @param fatal_failure_desc description of fatal failure
  @param fatal_failure_desc_len failure description len, failu

  @return Return 0 if access is granted, otherwise an appropri
  errno should be returned.
 */
typedef int mpo_vnode_check_signature_t(
    struct vnode *vp,
    struct label *label,
    struct cs_blob *cs_blob,
    unsigned int *cs_flags,

    int flags,
    char **fatal_failure_desc, size_t *fatal_failure_desc_len
);
```

G:\mac_policy\mac_policy_v52.h

```
/**
  @brief Access control check after determining the code directory hash
  @param vp vnode vnode to combine into proc
  @param label label associated with the vnode
  @param cs_blob the code signature to check
  @param cs_flags update code signing flags if needed
  @param signer_type output parameter for the code signature's signer type
  @param flags operational flag to mpo_vnode_check_signature
  @param fatal_failure_desc description of fatal failure
  @param fatal_failure_desc_len failure description len, failure is fatal if non-0

  @return Return 0 if access is granted, otherwise an appropriate value for
  errno should be returned.
 */
typedef int mpo_vnode_check_signature_t(
    struct vnode *vp,
    struct label *label,
    struct cs_blob *cs_blob,
    unsigned int *cs_flags,
    unsigned int *signer_type,
    int flags,
    char **fatal_failure_desc, size_t *fatal_failure_desc_len
);
```

Mandatory Access Control Policy (count)

CASE 3. Interfaces were inserted into the middle of the dispatch table

G:\mac_policy\mac_policy_v11.h

```
    mpo_proc_check_map_anon_t       *mpo_proc_check_map_anon;
    mpo_vnode_check_fsgetpath_t     *mpo_vnode_check_fsgetpath;
    mpo_iokit_check_open_t          *mpo_iokit_check_open;

    mpo_vnode_notify_rename_t       *mpo_vnode_notify_rename;
    mpo_reserved_hook_t         *mpo_reserved14;
    mpo_reserved_hook_t         *mpo_reserved15;
    mpo_reserved_hook_t         *mpo_reserved16;
    mpo_reserved_hook_t         *mpo_reserved17;
```

G:\mac_policy\mac_policy_v13_2050.7.9.h

```
    mpo_proc_check_map_anon_t       *mpo_proc_check_map_anon;
    mpo_vnode_check_fsgetpath_t     *mpo_vnode_check_fsgetpath;
    mpo_iokit_check_open_t          *mpo_iokit_check_open;
    mpo_proc_check_ledger_t         *mpo_proc_check_ledger;
    mpo_vnode_notify_rename_t       *mpo_vnode_notify_rename;
    mpo_thread_label_init_t         *mpo_thread_label_init;
    mpo_thread_label_destroy_t      *mpo_thread_label_destroy;
    mpo_system_check_kas_info_t *mpo_system_check_kas_info;
```

Mandatory Access Control Policy (count)

CASE 4. Interfaces have been rewritten but forgot to upgrade the policy version number

G:\mac_policy\mac_policy_v13_2050.7.9.h

```
    mpo_thread_label_init_t          *mpo_thread_label_init;
    mpo_thread_label_destroy_t       *mpo_thread_label_destroy;
    mpo_system_check_kas_info_t *mpo_system_check_kas_info;
    mpo_reserved_hook_t          *mpo_reserved18;
    mpo_reserved_hook_t          *mpo_reserved19;
    mpo_reserved_hook_t          *mpo_reserved20;
```

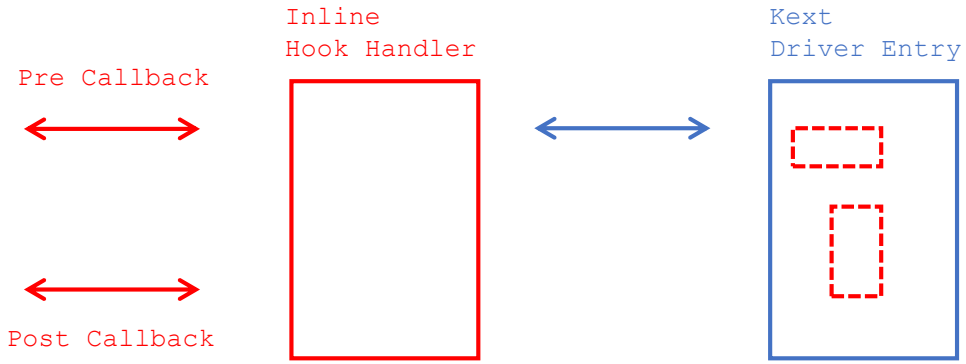
G:\mac_policy\mac_policy_v13_2050.24.15.h

```
    mpo_thread_label_init_t          *mpo_thread_label_init;
    mpo_thread_label_destroy_t       *mpo_thread_label_destroy;
    mpo_system_check_kas_info_t *mpo_system_check_kas_info;
    mpo_reserved_hook_t          *mpo_reserved18;
    mpo_vnode_notify_open_t          *mpo_vnode_notify_open;
    mpo_reserved_hook_t          *mpo_reserved20;
```

Kemon: An Open-Source Pre and Post Callback-Based Framework

Why Kemon Framework?

```
(lldb) di -b -n OSKext::start
kernel.development`OSKext::start:
    0xffffff800ce1aa00 <+0>:    55              pushq  %rbp
    0xffffff800ce1aa01 <+1>:    48 89 e5        movq   %rsp, %rbp
    0xffffff800ce1aa04 <+4>:    41 57           pushq  %r15
    0xffffff800ce1aa06 <+6>:    41 56           pushq  %r14
    0xffffff800ce1aa08 <+8>:    41 55           pushq  %r13
    0xffffff800ce1aa0a <+10>:   41 54           pushq  %r12
    0xffffff800ce1aa0c <+12>:   53              pushq  %rbx
    0xffffff800ce1aa0d <+13>:   48 83 ec 28     subq   $0x28, %rsp
    0xffffff800ce1aa11 <+17>:   41 89 f6        movl   %esi, %r14d
    0xffffff800ce1aa14 <+20>:   49 89 ff        movq   %rdi, %r15
    0xffffff800ce1aa17 <+23>:   49 8b 07        movq   (%r15), %rax
    ...................
    0xffffff800ce1adfd <+1021>: 4c 8b 65 c0     movq   -0x40(%rbp), %r12
    0xffffff800ce1ae01 <+1025>: 49 8b 7f 48     movq   0x48(%r15), %rdi
    0xffffff800ce1ae05 <+1029>: 4c 89 e6        movq   %r12, %rsi
    0xffffff800ce1ae08 <+1032>: ff 55 b0        callq  *-0x50(%rbp)
    ...................
    0xffffff800ce1ae60 <+1120>: 5b              popq   %rbx
    0xffffff800ce1ae61 <+1121>: 41 5c           popq   %r12
    0xffffff800ce1ae63 <+1123>: 41 5d           popq   %r13
    0xffffff800ce1ae65 <+1125>: 41 5e           popq   %r14
    0xffffff800ce1ae67 <+1127>: 41 5f           popq   %r15
    0xffffff800ce1ae69 <+1129>: 5d              popq   %rbp
    0xffffff800ce1ae6a <+1130>: c3              retq
```



Kext Device Driver Monitoring and Blocking

```
[Kemon.kext] : action=MONITORING_KEXT_PRE_CALLBACK, uid=0, process(pid 59)=kextd, parent(ppid 1)=launchd, name=com.mandiant.monitor, path=/Applications/Monitor.app/Contents/PlugIns/monitor.kext, version=0.9.2…
[Kemon.kext] : Disassemble the OSKext::start(com.mandiant.monitor) -> startfunc(kmod_info, kmodStartData).
(02) ffd3 CALL RBX
(02) 89c3 MOV EBX, EAX
(02) 85db TEST EBX, EBX
[Kemon.kext] : In kext pre callback handler. Patching the driver entry point! name=com.mandiant.monitor, version=0.9.2, module base=0xffffff7f8e0cd000, module size=0x16000.
[Kemon.kext] : action=MONITORING_KEXT_POST_CALLBACK, uid=0, process(pid 59)=kextd, parent(ppid 1)=launchd, status=5, name=com.mandiant.monitor, version=0.9.2, module base=0xffffff7f8e0cd000, module size=0x160…
[Kemon.kext] : In kext post callback handler. status=5, name=com.mandiant.monitor, version=0.9.2, module base=0xffffff7f8e0cd000, module size=0x16000.
Kext com.mandiant.monitor start failed (result 0x5).
Kext com.mandiant.monitor failed to load (0xdc008017).
Failed to load kext com.mandiant.monitor (error 0xdc008017).
Failed to load /Applications/Monitor.app/Contents/PlugIns/monitor.kext - (libkern/kext) kext (kmod) start/stop routine failed.
```

kernel.development (kemon) Volatile

Subsystem: -- Category: -- Details 2018-08-01 17:55:36.647081

```
[Kemon.kext] : action=MONITORING_KEXT_PRE_CALLBACK, uid=0, process(pid 59)=kextd, parent(ppid 1)=launchd, name=com.mandiant.monitor, path=/Applications/Monitor.app/Contents/PlugIns/monitor.kext,
version=0.9.2, module base=0xffffff7f8e0cd000, module size=0x16000.
```

macOS Mandatory Access Control Policy Monitoring

```
[Kemon.kext] : macOS MAC policy[0]=AMFI(Apple Mobile File Integrity), load time flags=0(NULL), policy mpc=0xffffff7f89e234b8, policy ops=0xffffff7f89e22a40.
[Kemon.kext] :       handler address: 0xffffff7f89e1d234, module offset: com.apple.driver.AppleMobileFileIntegrity+0x5234, policy name: mpo_cred_check_label_update_execve.
[Kemon.kext] :       handler address: 0xffffff7f89e1d23f, module offset: com.apple.driver.AppleMobileFileIntegrity+0x523F, policy name: mpo_cred_label_associate.
[Kemon.kext] :       handler address: 0xffffff7f89e1d28c, module offset: com.apple.driver.AppleMobileFileIntegrity+0x528C, policy name: mpo_cred_label_destroy.
[Kemon.kext] :       handler address: 0xffffff7f89e1d31c, module offset: com.apple.driver.AppleMobileFileIntegrity+0x531C, policy name: mpo_cred_label_init.
[Kemon.kext] :       handler address: 0xffffff7f89e1bd72, module offset: com.apple.driver.AppleMobileFileIntegrity+0x3D72, policy name: mpo_cred_label_update_execve.
[Kemon.kext] :       handler address: 0xffffff7f89e1bb96, module offset: com.apple.driver.AppleMobileFileIntegrity+0x3B96, policy name: mpo_file_check_mmap.
[Kemon.kext] :       handler address: 0xffffff7f89e1dfcb, module offset: com.apple.driver.AppleMobileFileIntegrity+0x5FCB, policy name: mpo_file_check_library_validation.
[Kemon.kext] :       handler address: 0xffffff7f89e1e021, module offset: com.apple.driver.AppleMobileFileIntegrity+0x6021, policy name: mpo_policy_initbsd.
[Kemon.kext] :       handler address: 0xffffff7f89e1b776, module offset: com.apple.driver.AppleMobileFileIntegrity+0x3776, policy name: mpo_exc_action_check_exception_send.
[Kemon.kext] :       handler address: 0xffffff7f89e1b724, module offset: com.apple.driver.AppleMobileFileIntegrity+0x3724, policy name: mpo_exc_action_label_update.
[Kemon.kext] : macOS MAC policy[1]=Sandbox(Seatbelt sandbox policy), load time flags=0(NULL), policy mpc=0xffffff7f8a0e80c0, policy ops=0xffffff7f8a0e8118.
[Kemon.kext] :       handler address: 0xffffff7f8a0d2740, module offset: com.apple.security.sandbox+0x4740, policy name: mpo_cred_label_destroy.
[Kemon.kext] :       handler address: 0xffffff7f8a0d274c, module offset: com.apple.security.sandbox+0x474C, policy name: mpo_cred_label_update.
[Kemon.kext] :       handler address: 0xffffff7f8a0d27d1, module offset: com.apple.security.sandbox+0x47D1, policy name: mpo_file_check_mmap.
[Kemon.kext] :       handler address: 0xffffff7f8a0d289c, module offset: com.apple.security.sandbox+0x489C, policy name: mpo_mount_check_fsctl.
[Kemon.kext] :       handler address: 0xffffff7f8a0d28f9, module offset: com.apple.security.sandbox+0x48F9, policy name: mpo_mount_check_mount.
[Kemon.kext] :       handler address: 0xffffff7f8a0d2af9, module offset: com.apple.security.sandbox+0x4AF9, policy name: mpo_policy_init.
[Kemon.kext] :       handler address: 0xffffff7f8a0d2f1d, module offset: com.apple.security.sandbox+0x4F1D, policy name: mpo_policy_syscall.
[Kemon.kext] :       handler address: 0xffffff7f8a0d366a, module offset: com.apple.security.sandbox+0x566A, policy name: mpo_kext_check_query.
[Kemon.kext] :       handler address: 0xffffff7f8a0d399c, module offset: com.apple.security.sandbox+0x599C, policy name: mpo_iokit_check_nvram_delete.
[Kemon.kext] :       handler address: 0xffffff7f8a0d3a6d, module offset: com.apple.security.sandbox+0x5A6D, policy name: mpo_proc_check_set_host_special_port.
[Kemon.kext] :       handler address: 0xffffff7f8a0d3b1a, module offset: com.apple.security.sandbox+0x5B1A, policy name: mpo_vnode_check_trigger_resolve.
[Kemon.kext] :       handler address: 0xffffff7f8a0d3ca5, module offset: com.apple.security.sandbox+0x5CA5, policy name: mpo_posixsem_check_create.
```

macOS Mandatory Access Control Policy Blocking

```
[Kemon.kext] : In mac_policy_register callback handler. Blocking!
[Kemon.kext] : macOS MAC policy=procmon_m(procmon_m), load time flags=2(MPC_LOADTIME_FLAG_UNLOADOK), policy mpc=0xffffff7fa34fb198, policy ops=0xffffff7fa34fb1e8.
[Kemon.kext] :       handler address: 0xffffff7fa34f10bb, policy name: mpo_cred_label_update_execve.
[Kemon.kext] : In mac_policy_register callback handler. Blocking!
[Kemon.kext] : macOS MAC policy=dylibmon_m(dylibmon_m), load time flags=2(MPC_LOADTIME_FLAG_UNLOADOK), policy mpc=0xffffff7fa34fa6d0, policy ops=0xffffff7fa34fa720.
[Kemon.kext] :       handler address: 0xffffff7fa34edce5, policy name: mpo_file_check_mmap.
[Kemon.kext] : In mac_policy_register callback handler. Blocking!
[Kemon.kext] : macOS MAC policy=ttymon_grant_m(ttymon_grant_m), load time flags=2(MPC_LOADTIME_FLAG_UNLOADOK), policy mpc=0xffffff7fa34f90b0, policy ops=0xffffff7fa34f9150.
[Kemon.kext] :       handler address: 0xffffff7fa34eb6d1, policy name: mpo_pty_notify_grant.
[Kemon.kext] : In mac_policy_register callback handler. Blocking!
[Kemon.kext] : macOS MAC policy=ttymon_close_m(ttymon_close_m), load time flags=2(MPC_LOADTIME_FLAG_UNLOADOK), policy mpc=0xffffff7fa34f9100, policy ops=0xffffff7fa34f9bc8.
[Kemon.kext] :       handler address: 0xffffff7fa34ebcfe, policy name: mpo_pty_notify_close.
[Kemon.kext] : In mac_policy_register callback handler. Blocking!
[Kemon.kext] : macOS MAC policy=monitor_kextmon_m(monitor_kextmon_h), load time flags=2(MPC_LOADTIME_FLAG_UNLOADOK), policy mpc=0xffffff7fa34fbc60, policy ops=0xffffff7fa34fbcb0.
[Kemon.kext] :       handler address: 0xffffff7fa34f38ad, policy name: mpo_kext_check_load.
```

Other Features

File operation and process creation monitoring

Dynamic library monitoring

Network traffic monitoring, etc.

```
[Kemon.kext] : <UDP> localhost:10863->192.168.1.1:53.
                                  -*> MEMORY DUMP <*-
+--------------------+-------------------------------------------------+------------------+
|       ADDRESS      |  0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F | 0123456789ABCDEF |
| -------------------+-------------------------------------------------+----------------- |
|  0xffffff8049a116c0 | 9f 01 01 00 00 01 00 00  00 00 00 00 0c 73 61 66 | .............saf |
|  0xffffff8049a116d0 | 65 62 72 6f 77 73 69 6e  67 0a 67 6f 6f 67 6c 65 | ebrowsing.google |
|  0xffffff8049a116e0 | 61 70 69 73 03 63 6f 6d  00 00 01 00 01          | apis.com.....    |
+--------------------+-------------------------------------------------+------------------+
[Kemon.kext] : <DNS Query> localhost:10863->192.168.1.1:53, uid=501, process(pid 2612)=Google Chrome, parent(ppid 1)=launchd, query=safebrowsing.googleapis.com.
```

Try it Now!

https://github.com/didi/kemon

Q&A

wang yu

Didi Research America