Qualcomm

November 14th, 2023

Improving UAPI Compatibility Review with Automated Tooling

John Moon <quic_johmoo@quicinc.com>quic_johmoo@quicinc.com
Software Engineer
Qualcomm Innovation Center, Inc.

Agenda

1. UAPI Overview
2. How UAPI Compatibility is Maintained Today
3. Automated Tooling
4. Sysfs and Procfs Compatibility

UAPI Overview

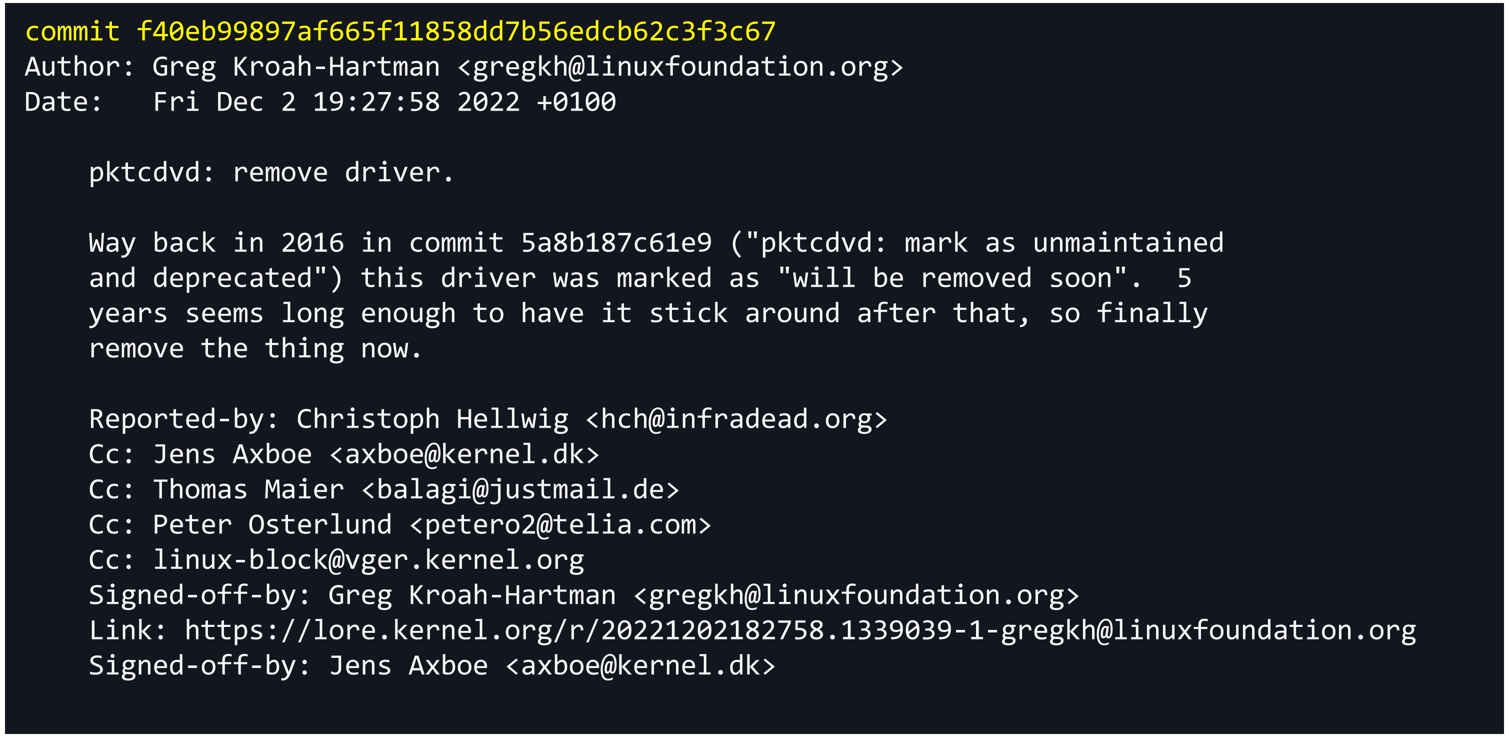
What is a UAPI?

- Userspace Application Programming Interface
- Basically, any interface between userspace and the kernel. Including:
  - System Calls
  - UAPI data structures (used in IOCTLs)
  - Module parameters
  - Sysfs files
  - Procfs files
  - … others?
- If it’s something that can change in the kernel and break compatibility with a userspace program, we consider it a UAPI.

```c
/**
 * struct gpioevent_request - Information about a GPIO event request
 * @lineoffset: the desired line to subscribe to events from, specified by
 * offset index for the associated GPIO device
 * @handleflags: desired handle flags for the desired GPIO line, such as
 * %GPIOHANDLE_REQUEST_ACTIVE_LOW or %GPIOHANDLE_REQUEST_OPEN_DRAIN
 * @eventflags: desired flags for the desired GPIO event line, such as
 * %GPIOEVENT_REQUEST_RISING_EDGE or %GPIOEVENT_REQUEST_FALLING_EDGE
 * @consumer_label: a desired consumer label for the selected GPIO line(s)
 * such as "my-listener"
 * @fd: if successful this field will contain a valid anonymous file handle
 * after a %GPIO_GET_LINEEVENT_IOCTL operation, zero or negative value
 * means error
 *
 * Note: This struct is part of ABI v1 and is deprecated.
 * Use &struct gpio_v2_line_request instead.
 */
struct gpioevent_request {
	__u32 lineoffset;
	__u32 handleflags;
	__u32 eventflags;
	char consumer_label[GPIO_MAX_NAME_SIZE];
	int fd;
};
```

`include/uapi/linux/gpio.h`

Linux doesn’t break UAPIs (intentionally)



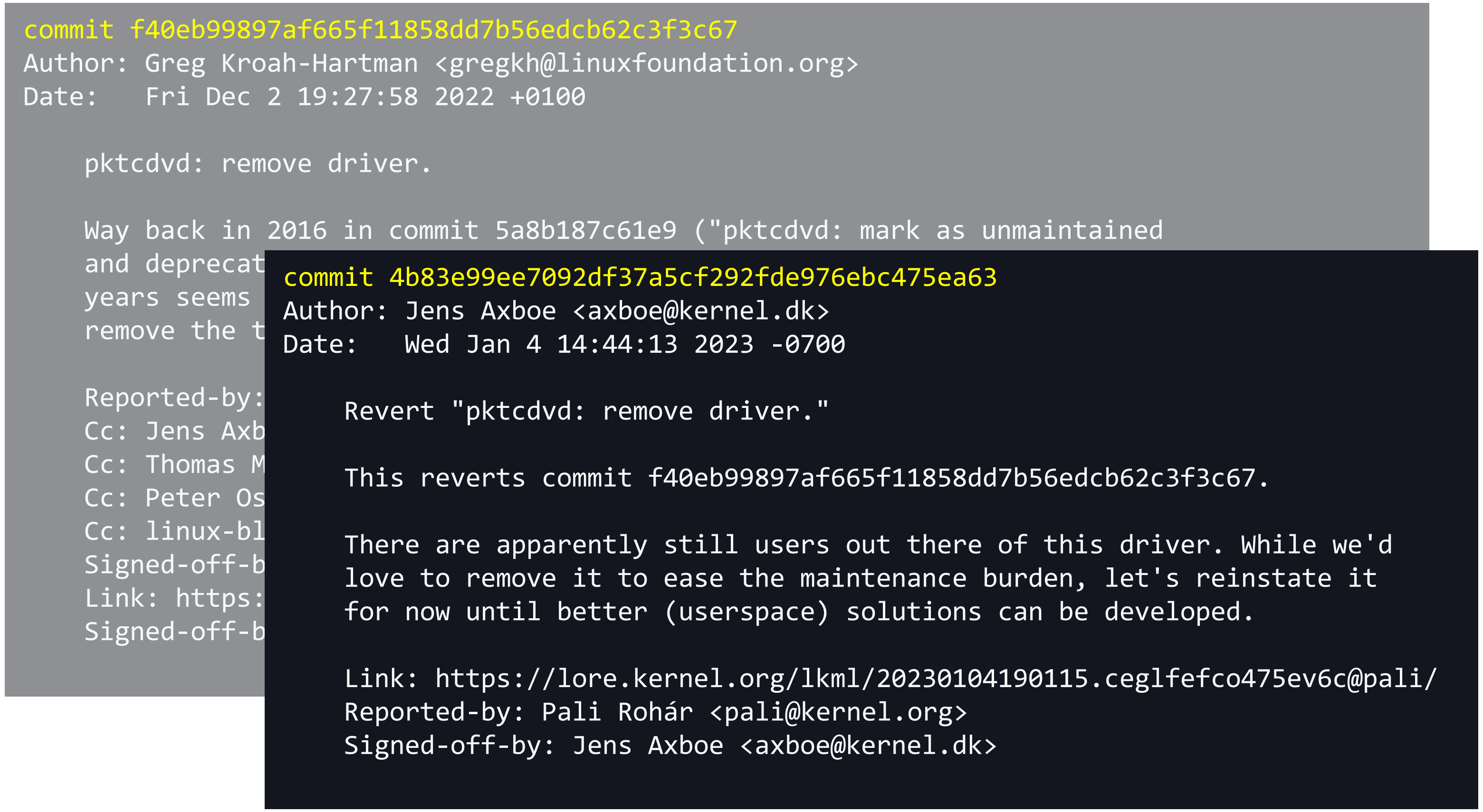
```
commit f40eb99897af665f11858dd7b56edcb62c3f3c67
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri Dec 2 19:27:58 2022 +0100

    pktcdvd: remove driver.

    Way back in 2016 in commit 5a8b187c61e9 ("pktcdvd: mark as unmaintained
    and deprecated") this driver was marked as "will be removed soon".  5
    years seems long enough to have it stick around after that, so finally
    remove the thing now.

    Reported-by: Christoph Hellwig <hch@infradead.org>
    Cc: Jens Axboe <axboe@kernel.dk>
    Cc: Thomas Maier <balagi@justmail.de>
    Cc: Peter Osterlund <petero2@telia.com>
    Cc: linux-block@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Link: https://lore.kernel.org/r/20221202182758.1339039-1-gregkh@linuxfoundation.org
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
```

Sometimes, if nobody is using a UAPI, it will get cleaned up…

“Breaking user programs simply isn't acceptable… We know that people use old [user space] binaries for years and years, and that making a new [kernel] release doesn't mean that you can just throw that out. **You can trust us.”**

https://lore.kernel.org/all/Pine.LNX.4.64.0601041356200.3668@g5.osdl.org/T/#u

Linux doesn’t break UAPIs (intentionally)



```
commit f40eb99897af665f11858dd7b56edcb62c3f3c67
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri Dec 2 19:27:58 2022 +0100

    pktcdvd: remove driver.

    Way back in 2016 in commit 5a8b187c61e9 ("pktcdvd: mark as unmaintained
    and deprecat
    years seems
    remove the t

    Reported-by:
    Cc: Jens Axb
    Cc: Thomas M
    Cc: Peter Os
    Cc: linux-bl
    Signed-off-b
    Link: https:
    Signed-off-b
```

```
commit 4b83e99ee7092df37a5cf292fde976ebc475ea63
Author: Jens Axboe <axboe@kernel.dk>
Date:   Wed Jan 4 14:44:13 2023 -0700

    Revert "pktcdvd: remove driver."

    This reverts commit f40eb99897af665f11858dd7b56edcb62c3f3c67.

    There are apparently still users out there of this driver. While we'd
    love to remove it to ease the maintenance burden, let's reinstate it
    for now until better (userspace) solutions can be developed.

    Link: https://lore.kernel.org/lkml/20230104190115.ceglfefco475ev6c@pali/
    Reported-by: Pali Rohár <pali@kernel.org>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
```

…but if it turns out there’s still a userspace program out there that needs it, it’ll get fixed.

“Breaking user programs simply isn't acceptable… We know that people use old [user space] binaries for years and years, and that making a new [kernel] release doesn't mean that you can just throw that out. **You can trust us.”**

https://lore.kernel.org/all/Pine.LNX.4.64.0601041356200.3668@g5.osdl.org/T/#u

How UAPI Compatibility is Maintained Today

How are UAPIs Kept Stable Today?

- Code review
- Testing

How are UAPIs Kept Stable Today?

- Code review
- Testing
- Automated tooling ← **This is what we’re trying to bring to the table.**

Automated Tooling

check-uapi.sh

A script to help check if a patch will break UAPI header compatibility

- Attempting to upstream now
  - As of this writing, v6 is the latest: https://lore.kernel.org/all/20231027193016.27516-1-quic_johmoo@quicinc.com/T/#t

```
check-uapi.sh - check for UAPI header stability across Git commits

By default, the script will check to make sure the latest commit (or current dirty changes) did
not introduce ABI changes when compared to HEAD^1. You can check against additional commit ranges
with the -b and -p options.

The script will not check UAPI headers for architectures other than the one defined in ARCH.

Usage: check-uapi.sh [-b BASE_REF] [-p PAST_REF] [-j N] [-l ERROR_LOG] [-i] [-q] [-v]

Options:
    -b BASE_REF    Base git reference to use for comparison. If unspecified or empty,
                   will use any dirty changes in tree to UAPI files. If there are no
                   dirty changes, HEAD will be used.
    -p PAST_REF    Compare BASE_REF to PAST_REF (e.g. -p v6.1). If unspecified or empty,
                   will use BASE_REF^1. Must be an ancestor of BASE_REF. Only headers
                   that exist on PAST_REF will be checked for compatibility.
    -j JOBS        Number of checks to run in parallel (default: number of CPU cores).
    -l ERROR_LOG   Write error log to file (default: no error log is generated).
    -i             Ignore ambiguous changes that may or may not break UAPI compatibility.
    -q             Quiet operation.
    -v             Verbose operation (print more information about each header being checked).
```

`check-uapi.sh`

How does it work?

- Installs all UAPI headers before/after patch (`make headers_install`)

check-uapi.sh

How does it work?

- Installs all UAPI headers before/after patch (**`make headers_install`**)
- Runs **`abidiff`** (a libabigail command line tool) on every UAPI header

```
--- a/my_uapi_header.h
+++ b/my_uapi_header.h
@@ -1,4 +1,5 @@
 struct foo {
        __u32 x;
+       __u32 z;
        __u32 y;
 }
```

check-uapi.sh

How does it work?

- Installs all UAPI headers before/after patch (**`make headers_install`**)
- Runs **`abidiff`** (a libabigail command line tool) on every UAPI header
- **`abidiff`** reports any incompatibilities.

```
--- a/my_uapi_header.h
+++ b/my_uapi_header.h
@@ -1,4 +1,5 @@
 struct foo {
        __u32 x;
+       __u32 z;
        __u32 y;
 }
```

```
[C] 'struct foo' changed:
   type size changed from 64 to 96 (in bits)
   1 data member insertion:
     '__u32 z', at offset 32 (in bits)
   1 data member change:
     '__u32 y' offset changed from 32 to 64 (in bits) (by +32 bits)
```

check-uapi.sh

How does it work?

- Installs all UAPI headers before/after patch (**`make headers_install`**)
- Runs **`abidiff`** (a libabigail command line tool) on every UAPI header
- **`abidiff`** reports any incompatibilities.
- By default, **`abidiff`** is overzealous with common kernel paradigms.

```
--- a/my_uapi_header.h
+++ b/my_uapi_header.h
@@ -1,5 +1,6 @@
 enum colors {
        RED,
        GREEN,
+       BLUE,
        COLOR_MAX,
 };
```

```
[C] 'enum colors' changed:
   type size hasn't changed
   1 enumerator insertion:
     'colors::BLUE' value '2'
   1 enumerator change:
     'colors::COLOR_MAX' from value '2' to '3'
```

check-uapi.sh

How does it work?

- Installs all UAPI headers before/after patch (**`make headers_install`**)
- Runs **`abidiff`** (a libabigail command line tool) on every UAPI header
- **`abidiff`** reports any incompatibilities.
- By default, **`abidiff`** is overzealous with common kernel paradigms.
- Can specify suppressions to filter out these kinds of things:

```
[suppress_type]
  type_kind = enum
  changed_enumerators_regexp = .*_MAX$
```

```
--- a/my_uapi_header.h
+++ b/my_uapi_header.h
@@ -1,5 +1,6 @@
 enum colors {
        RED,
        GREEN,
+       BLUE,
        COLOR_MAX,
 };
```

☑

check-uapi.sh

How does it work?

- Installs all UAPI headers before/after patch (**`make headers_install`**)
- Runs **`abidiff`** (a libabigail command line tool) on every UAPI header
- **`abidiff`** reports any incompatibilities.
- By default, **`abidiff`** is overzealous with common kernel paradigms.
- Can specify suppressions to filter out these kinds of things:

```
[suppress_type]
  type_kind = enum
  changed_enumerators_regexp = .*_MAX$
```

```
--- a/my_uapi_header.h
+++ b/my_uapi_header.h
@@ -1,5 +1,6 @@
 enum colors {
        RED,
        GREEN,
+       BLUE,
        COLOR_MAX,
 };
```

✓

Big thanks to Dodji Seketeli on the Libabigail team for helping us add the features we needed to filter out false positives in the kernel code!

Example #1

Adding a simple #define

```
--- a/include/uapi/linux/acct.h
+++ b/include/uapi/linux/acct.h
@@ -21,7 +21,9 @@
 #include <asm/param.h>
 #include <asm/byteorder.h>

+#define FOO
```

Example #1

Adding a simple #define

```
--- a/include/uapi/linux/acct.h
+++ b/include/uapi/linux/acct.h
@@ -21,7 +21,9 @@
 #include <asm/param.h>
 #include <asm/byteorder.h>

+#define FOO
```

```
% ./scripts/check-uapi.sh
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
All 912 UAPI headers compatible with x86 appear to be backwards compatible
```

Example #2

Changing a type

```
--- a/include/uapi/linux/bpf.h
+++ b/include/uapi/linux/bpf.h
@@ -74,7 +74,7 @@ struct bpf_insn {
        __u8    dst_reg:4;      /* dest register */
        __u8    src_reg:4;      /* source register */
        __s16   off;            /* signed offset */
-       __s32   imm;            /* signed immediate constant */
+       __u32   imm;            /* unsigned immediate constant */
 };
```

Example #2

Changing a type

```
--- a/include/uapi/linux/bpf.h
+++ b/include/uapi/linux/bpf.h
@@ -74,7 +74,7 @@ struct bpf_insn {
        __u8    dst_reg:4;      /* dest register */
        __u8    src_reg:4;      /* source register */
        __s16   off;            /* signed offset */
-       __s32   imm;            /* signed immediate constant */
+       __u32   imm;            /* unsigned immediate constant */
 };
```

```
% ./scripts/check-uapi.sh
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
==== ABI differences detected in include/linux/bpf.h from HEAD -> dirty tree ====
    [C] 'struct bpf_insn' changed:
      type size hasn't changed
      1 data member change:
        type of '__s32 imm' changed:
          typedef name changed from __s32 to __u32 at int-ll64.h:27:1
          underlying type 'int' changed:
            type name changed from 'int' to 'unsigned int'
            type size hasn't changed
==================================================================================

1/912 UAPI headers compatible with x86 appear _not_ to be backwards compatible
```

Example #2

Changing a type

```
--- a/include/uapi/linux/bpf.h
+++ b/include/uapi/linux/bpf.h
@@ -74,7 +74,7 @@ struct bpf_insn {
        __u8    dst_reg:4;      /* dest register */
        __u8    src_reg:4;      /* source register */
        __s16   off;            /* signed offset */
-       __s32   imm;            /* signed immediate constant */
+       __u32   imm;            /* unsigned immediate constant */
 };
```

```
% ./scripts/check-uapi.sh -i
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
All 912 UAPI headers compatible with x86 appear to be backwards compatible
```

Example #3

Re-ordering a member

```
--- a/include/uapi/linux/bpf.h
+++ b/include/uapi/linux/bpf.h
@@ -71,8 +71,8 @@ enum {

 struct bpf_insn {
        __u8    code;           /* opcode */
-       __u8    dst_reg:4;      /* dest register */
        __u8    src_reg:4;      /* source register */
+       __u8    dst_reg:4;      /* dest register */
        __s16   off;            /* signed offset */
        __s32   imm;            /* signed immediate constant */
 };
```

Example #3

Re-ordering a member

```
--- a/include/uapi/linux/bpf.h
+++ b/include/uapi/linux/bpf.h
@@ -71,8 +71,8 @@ enum {

 struct bpf_insn {
        __u8    code;           /* opcode */
-       __u8    dst_reg:4;      /* dest register */
        __u8    src_reg:4;      /* source register */
+       __u8    dst_reg:4;      /* dest register */
        __s16   off;            /* signed offset */
        __s32   imm;            /* signed immediate constant */
 };
```

```
% ./scripts/check-uapi.sh -i
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
==== ABI differences detected in include/linux/bpf.h from HEAD -> dirty tree ====
    [C] 'struct bpf_insn' changed:
      type size hasn't changed
      2 data member changes:
        '__u8 dst_reg' offset changed from 8 to 12 (in bits) (by +4 bits)
        '__u8 src_reg' offset changed from 12 to 8 (in bits) (by -4 bits)
====================================================================================

1/912 UAPI headers compatible with x86 appear _not_ to be backwards compatible
```

Example #4

Arch-specific headers

```
--- a/arch/arm64/include/uapi/asm/sigcontext.h
+++ b/arch/arm64/include/uapi/asm/sigcontext.h
@@ -70,6 +70,7 @@ struct sigcontext {
 struct _aarch64_ctx {
        __u32 magic;
        __u32 size;
+       __u32 new_var;
 };
```

Example #4

Arch-specific headers

```
--- a/arch/arm64/include/uapi/asm/sigcontext.h
+++ b/arch/arm64/include/uapi/asm/sigcontext.h
@@ -70,6 +70,7 @@ struct sigcontext {
 struct _aarch64_ctx {
        __u32 magic;
        __u32 size;
+       __u32 new_var;
 };
```

```
% ./scripts/check-uapi.sh
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
No changes to UAPI headers were applied between HEAD and dirty tree
```

Example #4

Arch-specific headers

```
--- a/arch/arm64/include/uapi/asm/sigcontext.h
+++ b/arch/arm64/include/uapi/asm/sigcontext.h
@@ -70,6 +70,7 @@ struct sigcontext {
 struct _aarch64_ctx {
        __u32 magic;
        __u32 size;
+       __u32 new_var;
 };
```

```
% CC=aarch64-linux-gnu-gcc ARCH=arm64 ./scripts/check-uapi.sh
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
==== ABI differences detected in include/asm/sigcontext.h from HEAD -> dirty tree =
    [C] 'struct _aarch64_ctx' changed:
      type size changed from 64 to 96 (in bits)
      1 data member insertion:
        '__u32 new_var', at offset 64 (in bits) at sigcontext.h:73:1
================================================================================

1/884 UAPI headers compatible with arm64 appear _not_ to be backwards compatible
```

Example #4

Arch-specific headers

```
--- a/arch/arm64/include/uapi/asm/sigcontext.h
+++ b/arch/arm64/include/uapi/asm/sigcontext.h
@@ -70,6 +70,7 @@ struct sigcontext {
 struct _aarch64_ctx {
        __u32 magic;
        __u32 size;
+       __u32 new_var;
 };
```

```
% CC=aarch64-linux-gnu-gcc ARCH=arm64 ./scripts/check-uapi.sh -i
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
All 884 UAPI headers compatible with arm64 appear to be backwards compatible
```

- Is this structure expansion an ambiguous break?
- Depending on how the struct expansion is handled in kernelspace, this could be fine.
- Userspace and kernelspace will disagree on the size of the struct.
- When using **`copy_struct_from_user()`** and the **`_IOC_SIZE`** macro, the structure can be zero-extended/truncated to deal with the disagreement.

Example #5

Cross-dependencies

```
--- a/include/uapi/linux/types.h
+++ b/include/uapi/linux/types.h
@@ -56,7 +56,7 @@ typedef __u32 __bitwise __wsum;
 #define __aligned_be64 __be64 __attribute__((aligned(8)))
 #define __aligned_le64 __le64 __attribute__((aligned(8)))

-typedef unsigned __bitwise __poll_t;
+typedef unsigned short __bitwise __poll_t;
```

Example #5

Cross-dependencies

```
--- a/include/uapi/linux/types.h
+++ b/include/uapi/linux/types.h
@@ -56,7 +56,7 @@ typedef __u32 __bitwise __wsum;
 #define __aligned_be64 __be64 __attribute__((aligned(8)))
 #define __aligned_le64 __le64 __attribute__((aligned(8)))

-typedef unsigned __bitwise __poll_t;
+typedef unsigned short __bitwise __poll_t;
```

```
% ./scripts/check-uapi.sh
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
==== ABI differences detected in include/linux/eventpoll.h from HEAD -> dirty tree =
    [C] 'struct epoll_event' changed:
      type size changed from 96 to 80 (in bits)
      2 data member changes:
        type of '__poll_t events' changed:
          underlying type 'unsigned int' changed:
            type name changed from 'unsigned int' to 'unsigned short int'
            type size changed from 32 to 16 (in bits)
        '__u64 data' offset changed from 32 to 16 (in bits) (by -16 bits)
===================================================================================
include/linux/eventpoll.h did not change between HEAD and dirty tree...
It's possible a change to one of the headers it includes caused this error:
#include <linux/fcntl.h>
#include <linux/types.h>

1/912 UAPI headers compatible with x86 appear _not_ to be backwards compatible
```

Example #6

UAPI Header Removals

```
--- a/include/uapi/asm-generic/Kbuild
+++ b/include/uapi/asm-generic/Kbuild
@@ -31,6 +31,6 @@ mandatory-y += stat.h
 mandatory-y += statfs.h
 mandatory-y += swab.h
 mandatory-y += termbits.h
-mandatory-y += termios.h
+# mandatory-y += termios.h
 mandatory-y += types.h
 mandatory-y += unistd.h
```

Example #6

UAPI Header Removals

```
--- a/include/uapi/asm-generic/Kbuild
+++ b/include/uapi/asm-generic/Kbuild
@@ -31,6 +31,6 @@ mandatory-y += stat.h
 mandatory-y += statfs.h
 mandatory-y += swab.h
 mandatory-y += termbits.h
-mandatory-y += termios.h
+# mandatory-y += termios.h
 mandatory-y += types.h
 mandatory-y += unistd.h
```

```
% ./scripts/check-uapi.sh
Installing user-facing UAPI headers from dirty tree... OK
Installing user-facing UAPI headers from HEAD... OK
Checking changes to UAPI headers between HEAD and dirty tree...
==== UAPI header include/asm/termios.h was removed between HEAD and dirty tree ====

1/912 UAPI headers compatible with x86 appear _not_ to be backwards compatible
```

Example #7

Checking swaths of history

```
% ./scripts/check-uapi.sh -b v6.1 -p v6.0
Installing user-facing UAPI headers from v6.1... OK
Installing user-facing UAPI headers from v6.0... OK
Checking changes to UAPI headers between v6.0 and v6.1...

--- snip ---
37/907 UAPI headers compatible with x86 appear _not_ to be backwards compatible
```

Example #7

Checking swaths of history

```
% ./scripts/check-uapi.sh -b v6.1 -p v6.0
Installing user-facing UAPI headers from v6.1... OK
Installing user-facing UAPI headers from v6.0... OK
Checking changes to UAPI headers between v6.0 and v6.1...

--- snip ---
37/907 UAPI headers compatible with x86 appear _not_ to be backwards compatible
```

- The script makes no assumptions about the intent of the author.
  - Intentional breakages are still flagged.
- Refactors which don’t impact userspace may be flagged.
  - If a file is moved, that counts as a “removal.”

check-module-params.sh

A script to help check if a patch will break module parameter compatibility

- Attempting to upstream now
  - As of this writing, v6 is the latest: https://lore.kernel.org/all/20231027193016.27516-1-quic_johmoo@quicinc.com/T/#t

```
check-module-params.sh - check for module parameter stability across git commits.

By default, the script will check to make sure the latest commit (or current
dirty changes) did not introduce changes when compared to HEAD^1. You can
check against additional commit ranges with the -b and -p options.

Usage: check-module-params.sh [-b BASE_REF] [-p PAST_REF] [-j N] [-l ERROR_LOG] [-q] [-v]

Options:
    -b BASE_REF    Base git reference to use for comparison. If unspecified or empty,
                   will use any dirty changes in tree to UAPI files. If there are no
                   dirty changes, HEAD will be used.
    -p PAST_REF    Compare BASE_REF to PAST_REF (e.g. -p v6.1). If unspecified or empty,
                   will use BASE_REF^1. Must be an ancestor of BASE_REF. Only headers
                   that exist on PAST_REF will be checked for compatibility.
    -j JOBS        Number of checks to run in parallel (default: number of CPU cores).
    -l ERROR_LOG   Write error log to file (default: no error log is generated).
    -q             Quiet operation (suppress stdout, still print stderr).
    -v             Verbose operation (print more information about each header being checked).
```

check-module-params.sh

A script to help check if a patch will break module parameter compatibility

- **Very** simple algorithm:
  1. Step through every changed file.
  2. Match all **`module_param.*(<args>)`** lines.
  3. Flag any changes.

```
--- a/net/mac80211/mlme.c
+++ b/net/mac80211/mlme.c
@@ -44,7 +44,7 @@
 #define IEEE80211_ASSOC_MAX_TRIES      3

 static int max_nullfunc_tries = 2;
-module_param(max_nullfunc_tries, int, 0644);
+module_param(max_nullfunc_tries, int, 0600);
 MODULE_PARM_DESC(max_nullfunc_tries,
                 "Maximum nullfunc tx tries ...
```

check-module-params.sh

A script to help check if a patch will break module parameter compatibility

- **Very** simple algorithm:
  1. Step through every changed file
  2. Match all **`module_param.*(<args>)`** lines
  3. Flag any changes.

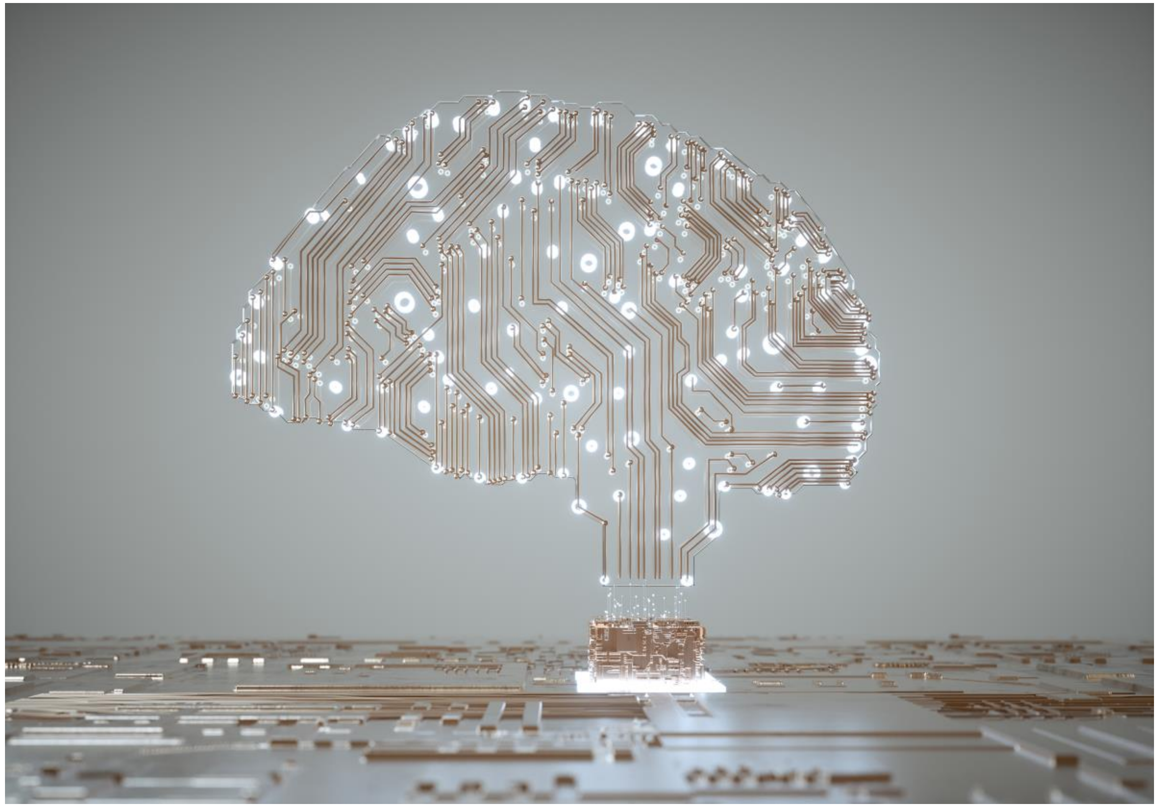
```
--- a/net/mac80211/mlme.c
+++ b/net/mac80211/mlme.c
@@ -44,7 +44,7 @@
 #define IEEE80211_ASSOC_MAX_TRIES      3

 static int max_nullfunc_tries = 2;
-module_param(max_nullfunc_tries, int, 0644);
+module_param(max_nullfunc_tries, int, 0600);
 MODULE_PARM_DESC(max_nullfunc_tries,
                 "Maximum nullfunc tx tries ...
```

```
% ./scripts/check-module-params.sh
Checking files between HEAD and dirty tree for module parameter compatibility...
Module parameter "max_nullfunc_tries" in net/mac80211/mlme.c changed!
  Original args: int,0644
       New args: int,0600


1/1 modified files with modules parameters appear _not_ to be backwards compatible
```

Sysfs and Procfs Compatibility

Sysfs and Procfs

- We currently don’t have any tooling to check sysfs and procfs compatibility.
- As a community, can we come up with a solution?



Idea #1

…to get the juices flowing

- We already have many sysfs and procfs interfaces documented in **`Documentation/ABI`**.
- The current format is machine-parsable, and there's already a script which does this (**`scripts/get_abi.pl`**).
- **`get_abi.pl`** already has support for checking if there are any undefined sysfs nodes on your system.

```
What:           /sys/class/leds/<led>/max_brightness
Date:           March 2006
KernelVersion:  2.6.17
Contact:        Richard Purdie <rpurdie@rpsys.net>
Description:

                Maximum brightness level for this LED, default is 255 (LED_FULL).

                If the LED does not support different brightness levels, this
                should be 1.
```

`Documentation/ABI/testing/sysfs-class-led`

Idea #1

…to get the juices flowing

- Could this documentation interface be extended (like device tree bindings)?
- We could have a generic test harness.
- Or, better yet, static analysis of code.

```
title: LED sysfs nodes
node:
  name_glob: /sys/class/leds/*/max_brightness
  date: March 2006
  kernel_version: "2.6.17"
  contact: Richard Purdie <rpurdie@rpsys.net>
  description:
    Maximum brightness level for this LED.
  type: u32
  default: 255
  min_value: 1
  max_value: 255
  null_value: 1
```

Documentation/ABI/testing/sysfs-class-led.yaml (?)

```
fn test_u32(node: &Node) {
    assert_eq!(get_val(node), node["default"]);
    set_val(node, node_val["min_val"]);
    assert_eq!(get_val(node), node["min_val"]);
    // ...
}

fn run_test(node: &Node) {
    match node.get("type") {
        "u32" => test_u32(node),
        _ => unimplemented!()
    }
}

fn main() {
    for node in running_sysfs() {
        if documented_nodes().contains_key(node) {
            run_test(node);
        } else {
            eprintln!("{} undocumented!", node);
        }
    }
}
```

Idea #1

…to get the juices flowing

- For live testing, we can’t just go poking values into every sysfs node.

```
What:           /sys/power/state
Date:           November 2016
Contact:        Rafael J. Wysocki <rjw@rjwysocki.net>
Description:
                The /sys/power/state file controls system sleep states.
                Reading from this file returns the available sleep state
                labels, which may be "mem" (suspend), "standby" (power-on
                suspend), "freeze" (suspend-to-idle) and "disk" (hibernation).

                Writing one of the above strings to this file causes the system
                to transition into the corresponding state, if available.

                See Documentation/admin-guide/pm/sleep-states.rst for more
                information.
```

`Documentation/ABI/testing/sysfs-power`

Questions? Ideas? Suggestions?

Thank you

Qualcomm

Follow us on:

For more information, visit us at:
qualcomm.com & qualcomm.com/blog

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

References in this presentation to "Qualcomm" may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patented technologies are licensed by Qualcomm Incorporated.