author	Gabriel Shahrouzi <gshahrouzi@gmail.com>	2025-04-05 16:30:36 -0400
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2025-05-02 07:50:58 +0200
commit	1fe9b92eede32574dbe05b5bdb6ad666b350bed0 (patch)	
tree	afab08c431fc91a794cff0d65ab73cf1fdd75b1f	
parent	eee189ccd46b46e4a4b9d787cd65dda5025549e8 (diff)	
download	linux-1fe9b92eede32574dbe05b5bdb6ad666b350bed0.tar.gz	

perf/core: Fix WARN_ON(!ctx) in __free_event() for partial init

```
[ Upstream commit 0ba3a4ab76fd3367b9cb680cad70182c896c795c ]

Move the get_ctx(child_ctx) call and the child_event->ctx assignment to
occur immediately after the child event is allocated. Ensure that
child_event->ctx is non-NULL before any subsequent error path within
inherit_event calls free_event(), satisfying the assumptions of the
cleanup code.

Details:

There's no clear Fixes tag, because this bug is a side-effect of
multiple interacting commits over time (up to 15 years old), not
a single regression.

The code initially incremented refcount then assigned context
immediately after the child_event was created. Later, an early
validity check for child_event was added before the
refcount/assignment. Even later, a WARN_ON_ONCE() cleanup check was
added, assuming event->ctx is valid if the pmu_ctx is valid.
The problem is that the WARN_ON_ONCE() could trigger after the initial
check passed but before child_event->ctx was assigned, violating its
precondition. The solution is to assign child_event->ctx right after
its initial validation. This ensures the context exists for any
subsequent checks or cleanup routines, resolving the WARN_ON_ONCE().

To resolve it, defer the refcount update and child_event->ctx assignment
directly after child_event->pmu_ctx is set but before checking if the
parent event is orphaned. The cleanup routine depends on
event->pmu_ctx being non-NULL before it verifies event->ctx is
non-NULL. This also maintains the author's original intent of passing
in child_ctx to find_get_pmu_context before its refcount/assignment.

[ mingo: Expanded the changelog from another email by Gabriel Shahrouzi. ]

Reported-by: syzbot+ff3aa851d46ab82953a3@syzkaller.appspotmail.com
Signed-off-by: Gabriel Shahrouzi <gshahrouzi@gmail.com>
Signed-off-by: Ingo Molnar <mingo@kernel.org>
Cc: Peter Zijlstra <peterz@infradead.org>
Cc: Ravi Bangoria <ravi.bangoria@amd.com>
Cc: Kan Liang <kan.liang@linux.intel.com>
Cc: Oleg Nesterov <oleg@redhat.com>
```

Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
Link: https://lore.kernel.org/r/20250405203036.582721-1-gshahrouzi@gmail.com
Closes: https://syzkaller.appspot.com/bug?extid=ff3aa851d46ab82953a3
Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r-- kernel/events/core.c 6

1 files changed, 3 insertions, 3 deletions

```
diff --git a/kernel/events/core.c b/kernel/events/core.c
index b710976fb01b17..987807b1040ae0 100644
--- a/kernel/events/core.c
+++ b/kernel/events/core.c
@@ -13419,6 +13419,9 @@ inherit_event(struct perf_event *parent_event,
        if (IS_ERR(child_event))
                return child_event;

+       get_ctx(child_ctx);
+       child_event->ctx = child_ctx;
+
        pmu_ctx = find_get_pmu_context(child_event->pmu, child_ctx, child_event);
        if (IS_ERR(pmu_ctx)) {
                free_event(child_event);
@@ -13441,8 +13444,6 @@ inherit_event(struct perf_event *parent_event,
                return NULL;
        }

-       get_ctx(child_ctx);
-
        /*
         * Make the child state follow the state of the parent event,
         * not its attr.disabled bit.  We hold the parent's mutex,
@@ -13463,7 +13464,6 @@ inherit_event(struct perf_event *parent_event,
                local64_set(&hwc->period_left, sample_period);
        }

-       child_event->ctx = child_ctx;
        child_event->overflow_handler = parent_event->overflow_handler;
        child_event->overflow_handler_context
                = parent_event->overflow_handler_context;
```