Acen28 / CVE-2026-26399-Disclosure Public

Code Issues Pull requests Actions Projects Security and quality

1 Branch 0 Tags Go to file Go to file Code

Acen28 Document CVE-2026-26399 in README 197e977 · last week

README.md Document CVE-2026-2639… last week

README

CVE-2026-26399: Stack-based Use-After-Return in Arduino_Core_STM32 (Legacy Versions)

Summary

A stack-based use-after-return vulnerability exists in legacy versions of `Arduino_Core_STM32` prior to version `1.7.0`.

The issue is located in the `pwm_start()` implementation, where a `TIM_HandleTypeDef` object is allocated on the stack and its address is passed into HAL initialization logic. That pointer may later be retained in a global timer handle registry and dereferenced asynchronously by interrupt service routines after the original stack frame has already returned.

This vulnerability has been assigned **CVE-2026-26399** by MITRE.

Affected Product

- Product: `Arduino_Core_STM32`
- Vendor / Project: `stm32duino` / STMicroelectronics ecosystem
- Affected versions: `v0.1.0` through `v1.6.1`

- Fixed version: `v1.7.0`

Vulnerability Type

- CWE-562: Use After Return
- Stack-based dangling pointer / use-after-return

Technical Details

In the legacy implementation of `pwm_start()`, a `TIM_HandleTypeDef` structure is created as a local stack variable:

```
void pwm_start(...) {
    TIM_HandleTypeDef timHandle = {};
    HAL_TIM_PWM_Init(&timHandle);
    ...
}
```

The address of this local object is passed into HAL support logic and may be stored in global timer-management state, such as a global timer handle registry.

After `pwm_start()` returns, the stack object is no longer valid. However, subsequent asynchronous timer interrupt handlers may still dereference the stale pointer, causing memory corruption and undefined behavior.

Impact

Possible security impact includes:

- Denial of Service (for example, crash or HardFault)
- Potential code execution under favorable memory-layout and control conditions

Because this is a library-level issue, real exploitability depends on how the vulnerable library is integrated into firmware and how attacker-controlled input can influence stack reuse and interrupt timing.

Affected Components

- `cores/arduino/stm32/analog.c`
- `pwm_start()`

- `TIM_HandleTypeDef`
- global timer handle tracking logic (for example, `timer_handles`)

Fix

The issue appears to have been resolved indirectly in version `1.7.0` during a refactoring that replaced the previous stack-based handling path with a different timer-management design. No dedicated security advisory appears to have been issued for the legacy vulnerability.

Discovery

This issue was identified during firmware security research involving fuzzing and root-cause analysis of legacy STM32-based firmware built on `Arduino_Core_STM32`.

Vendor Contact

The vendor PSIRT was contacted. The response indicated that current versions are not affected because the issue had already been fixed in version `1.7.0`, but no dedicated CVE appears to have been requested by the vendor at that time.

Credit

Discovered by **Jin Chang**.

References

- Project repository: https://github.com/stm32duino/Arduino_Core_STM32
- CVE ID: `CVE-2026-26399`

Releases

No releases published

Packages

No packages published

Contributors 1

Acen28