Process High-Impact clang-tidy and Clang Static Analyzer Requests

CppAlliance Fellowship 2026

Peiqi Li

Mentors: Vassil Vassilev · Aaron Jomy · Martin Vassilev

Background and goal

- **The tools.** clang-tidy reports diagnostics and fix-its in CI and editors. The Clang Static Analyzer models program state to find leaks and null dereferences.
- **The problem.** Both trackers have a backlog of false positives, false negatives, unsafe fix-its, and performance issues. Noisy diagnostics lead users to suppress checks.
- **The goal.** Fix the highest-impact cases conservatively: no new false positives, no broken workflows, no higher analysis cost.

Pull requests

9 merged · 3 in review

PR	What it fixes	Component	Status
#199189	Keep typedefs inside extern "C" (IgnoreExternC on by default)	modernize-use-using	merged
#199905	FP on std::forward of derived-class members in move constructors	bugprone-use-after-move	merged
#198085	FP when comparing values through distinct type aliases	misc-redundant-expression	merged
#208682	mergeFixits() moved to clang::edit and reused by clangd	clangd + clang-tidy	merged
#220548	FP on empty initializer lists (syntactic vs semantic form)	readability-trailing-comma	merged
#216336	Query-based custom checks enabled in clangd	clangd	merged
#146	Documentation for ExperimentalCustomChecks	clangd-www	merged
#225585	FP on C++20 ordering comparisons after libstdc++ 16 rename	modernize-use-nullptr	merged
#227984	No fix-it for one-character strings from macro expansions	performance-prefer-single-char-overloads	merged
#221587	FN: pointers stored in heap fields treated as escaped	analyzer · NewDeleteLeaks	in review
#225048	FP: LocAsInteger lost through identity operations	analyzer · unix.Malloc	in review
#231736	FP after memset(p, 0, sizeof(*p))	analyzer · CStringChecker	in review

8 patches in llvm/llvm-project, 1 in llvm/clangd-www.

Representative clang-tidy fixes

- **modernize-use-using (#199189).** typedefs inside extern "C" blocks were rewritten to using, which produces invalid C in mixed-language headers. IgnoreExternC now defaults to true.

- **modernize-use-nullptr (#225585).** libstdc++ 16 renamed an internal type, so every C++20 spaceship comparison started warning. The new name is added to IgnoredTypes.

- **performance-prefer-single-char-overloads (#227984).** a fix-it on a one-character string from a macro replaced the macro with a raw literal. The fix-it is now suppressed; the warning remains.

Work done in clangd

- **Custom checks (#216336).** Checks written as clang-query matchers in .clang-tidy now produce diagnostics in clangd.
- **Opt-in.** Off by default; runs only when FastCheckFilter is Loose or None, not under Strict. Documented on clangd.website (#146).
- **Fix-it safety (#208682).** clangd now uses the shared mergeFixits() logic from clang::edit.
- **LineFilter (#202575).** Implemented, but review showed the request was a workaround. The real need is filtering by macro definition location; this is future work.

```
# clangd config
Diagnostics:
  ClangTidy:
    ExperimentalCustomChecks: true
    FastCheckFilter: Loose
```

Work done in the Clang Static Analyzer

These patches change how a checker models program state, not only which AST pattern it matches. All three are in review.

Checker	PR	Type	Change
NewDeleteLeaks	#221587	FN	Pointers stored in heap fields were treated as escaped, so real leaks were not reported.
unix.Malloc	#225048	FP	LocAsInteger was lost through identity operations, so the allocation looked leaked.
CStringChecker	#231736	FP	memset(p, 0, sizeof(*p)) invalidated *p instead of modelling the zeroing.

Modelling memset (#231736)

```
struct container { int *head; };

void test(struct container *c) {
  int *tail = 0;
  memset(c, 0, sizeof(*c));
  // extent is only known >= sizeof(*c)
  clang_analyzer_eval(c->head == 0);
  // before: TRUE and FALSE
  // after:  TRUE
  if (c->head != 0)
    *tail = 1;   // null-dereference FP
}
```

Root cause

c is a pointer parameter, so the extent is only known to be at least sizeof(*c). The checker required an exact match, fell back to invalidation, and discarded everything it knew about *c.

Fix

When the fill is zero and the length provably covers the pointee, bind a zeroed value instead of invalidating. Non-zero fills and symbolic extents keep the old behavior.

What's done and what's next

Done

- 9 PRs merged upstream: 8 in llvm/llvm-project, 1 in llvm/clangd-www.
- 6 clang-tidy checks corrected; clangd applies fix-its safely and supports opt-in custom checks.
- 3 analyzer patches in review. When merged, they report silent heap-field leaks and remove false leak/null-dereference reports.

Next

- Land the three analyzer patches (#221587, #225048, #231736).
- Write the ExcludeMacroExpansionRegex RFC from the LineFilter review (#202575).

How AI was used

Find the code

Use Cursor to locate where a check makes a decision.

Compare fixes

Use targeted prompts to compare possible fixes and trade-offs.

Test variations

Generate edge cases and regression test variants.

Review support

Check the change against tests and reviewer comments.

Thank you.