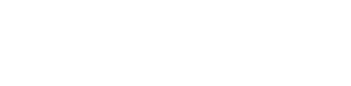


## https://godbolt.org/z/Gbh75aadj

```
struct node
    node* prev;
    node* next;
static_assert(! is_send_v<node>);
```

```
struct type
   type()
        auto n = std::make_shared<node>();
        [[maybe_unused]] auto this_capturing = [this] { run(); };
        static_assert(! is_send_v<decltype(this_capturing)>);
        [[maybe_unused]] auto this_n_capturing = [this, n] { run(); };
        static_assert(! is_send_v<decltype(this_n_capturing)>);
        [[maybe_unused]] auto n_ref_capturing = [&n] {};
        static assert(! is send v<decltype(n ref capturing)>);
        [[maybe_unused]] auto n_val_capturing = [n] {};
        static assert(! is send v<decltype(n val capturing)>);
   void run() {}
```

```
[[[maybe_unused]] auto non_capturing = [] (int) {};
static assert(is send v<decltype(non capturing)>);
int i = 0;
static assert(is_send_v<decltype(val_capturing)>);
static_assert(! is_send_v<decltype(ref_capturing)>);
```



```
[[maybe unused]] auto non capturing = [] (int) {};
struct node
                                         static_assert(is_send_v<decltype(non_capturing)>);
   node* prev;
                                         int i = 0:
   node* next;
                                         [[maybe_unused]] auto val_capturing = [i] (int) {};
};
                                         static_assert(is_send_v<decltype(val_capturing)>);
static_assert(! is_send_v<node>);
                                          [[maybe_unused]] auto ref_capturing = [&i] (int) {};
struct type
                                         static_assert(! is_send_v<decltype(ref_capturing)>);
    type()
        auto n = std::make_shared<node>();
        [[maybe_unused]] auto this_capturing = [this] { run(); };
        static_assert(! is_send_v<decltype(this_capturing)>);
        [[maybe_unused]] auto this_n_capturing = [this, n] { run(); };
        static_assert(! is_send_v<decltype(this_n_capturing)>);
        [[maybe_unused]] auto n_ref_capturing = [&n] {};
        static_assert(! is_send_v<decltype(n_ref_capturing)>);
        [[maybe_unused]] auto n_val_capturing = [n] {};
        static_assert(! is_send_v<decltype(n_val_capturing)>);
    void run() {}
```

**|};** 

## Problems: Summary

- <del>Nested pointers</del>
- this pointers
- Global leaked pointers
- Local leaked pointers