

Home

PUBLIC

Stack Overflow

Tags

Users

FIND A JOB

Jobs

Companies

TEAMS

What's this?

Free 30 Day Trial

Memory leak and Memory_free_on_stack variable in c++ while running in checkmarx

Asked 1 year, 3 months agoActive 1 year, 3 months agoViewed 252 times

I am trying to remediate the vulnerabilities that is deducted by checkmarx tool. Below is the sample code which is similar to my project that I am currently working on which is throwing memory leak and memory free on stack variables.

Memory is allocated in the called function and deleted in Main function.

If sum is allocated in the main function, there won't be any Memory leak or Memory free on stack variables. But I couldn't do the same in my actual code which involves some other variables to determine the size of the pointer variables before allocating.

```
void addition(int *a, int *b, int** sum)
{
    *sum = new int[2]; //MEMORY LEAK is thrown here

    if (*sum)
    {
        for (int i = 0; i < 2; i++)
        {
            cout << "Enter two numbers: " << endl;
            cin >> a[i] >> b[i];
            (*sum)[i] = a[i] + b[i];
            cout << "sum:" << (*sum)[i] << endl;
        }
    }
}

int main()
{
    int* p1 = NULL;
    p1=new int[2];
    int* p2 = NULL;
    p2 =new int[2];
    int *sum = NULL; //MemoryFree_On_StackVariable is thrown here

    addition(p1 ,p2, &sum);


    for (int i = 0; i < 2; i++)
    {
        cout << "sum is " << sum[i] << endl;
    }
    if (p1)
    {
        delete[] p1;
        p1 = NULL;
        cout << "p1 is deleted" <<endl;
    }
}
```

How could I rectify the vulnerability? are those vulnerability thrown rightly? should the memory be allocated and deleted in same function?

c++securitycheckmarx

share follow

asked Aug 9 '19 at 11:33

Indusha S

3102

- 4 Why don't you just use stl containers? Or at least wrap the sum type into a simple class, that frees the memory on destruction. Using raw new and delete is just bad style and will at some point leak memory. – MofX Aug 9 '19 at 11:39
- 3 I don't know what this tool you are using is about, but I can't see any leaks in the code above. It's perfectly OK to allocate memory in one function and free it in another. – john Aug 9 '19 at 11:58

add a comment

1 Answer

ActiveOldestVotes

To be honest, I don't know why in your case you have a memory leak because **everything seems correct** (to me at least).

But if you call addition() several times with the same sum pointer, you will create memory leaks for sure (except if you don't forget to delete[] sum; before each additional call to addition()) because you open the door to reallocate an already allocated memory which causes a memory leak.

I think this could be a reason to allocate/deallocate memory in the same scope. But in your case, I really don't understand.
I already had done similar things (delegate the deallocation task elsewhere) and I had no problems with it.

EDIT:


I don't know the checkmarx tool you are using. But it seems that it is not able to keep tracking pointers through function calls.
This is why it says that you try to free a stack variable in the main() function and why it says you don't delete an allocated variable in the addition() function.
This explains why it is expecting allocation/deallocation in the same block.

According to me, you don't have any error, but checkmarx is not working as it should do. Perhaps you can use Valgrind for your check instead.

share follow

edited Aug 9 '19 at 12:25

answered Aug 9 '19 at 12:01

Fareanor

3,5531324

add a comment

Your Answer

B

I

↺↻

💬


{ }


🖼


☰☱☲☳

↶↷

Sign up or log in

 Sign up using Google

 Sign up using Facebook

 Sign up using Email and Password

Post Your Answer

By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy

Not the answer you're looking for? Browse other questions tagged c++securitycheckmarx or ask your own question.


Post as a guest


Name

Email


Required, but never shown


The Overflow Blog


 How to write an effective developer resume: Advice from a hiring manager

 Podcast 290: This computer science degree is brought to you by Big Tech

Featured on Meta

 "Question closed" notifications experiment results and graduation










 MAINTENANCE WARNING: Possible downtime early morning Dec 2/4/9 UTC (8:30PM...)

 Congratulations VonC for reaching a million reputation

Related

- 3390 What are the differences between a pointer variable and a reference variable in C++?
- 1891 How can I profile C++ code running on Linux?
- 4 C++: memory leaks
- 1 C++ memory leak detecting method
- 1966 C++11 introduced a standardized memory model. What does it mean? And how is it going to affect C++ programming?
- 0 C++ dynamic memory leaks
- 0 Memory leak when create std::shared_ptr to assign member variable
- 1 Proper way to check for memory leaks in Visual C++
- 3 Memory leak on the stack
- 0 Searching for memory leak in C++ Builder XE4

Hot Network Questions

-  How to update Drupal custom module after changing .install file
-  How do I legally resign in Germany when no one is at the office?
-  Figuring out from a map which direction is downstream for a river?
-  Tower of Pisa function
-  What inner tube to buy for 47 tire size, 650bx47c tires?
-  What happens if my Zurich public transportation ticket expires while I am traveling?
-  How many pawns make up for a missing queen in the endgame?
-  What does the verb "to monograph" mean in documents context?
-  I got my money returned for a product that I did not return

more hot questions

Question feed



STACK OVERFLOW

- Questions
- Jobs
- Developer Jobs Directory
- Salary Calculator
- Help
- Mobile
- Disable Responsiveness

PRODUCTS

- Teams
- Talent
- Advertising
- Enterprise

COMPANY

- About
- Press
- Work Here
- Legal
- Privacy Policy
- Contact Us

STACK EXCHANGE NETWORK

- Technology
- Life / Arts
- Culture / Recreation
- Science
- Other

BlogFacebookTwitterLinkedInInstagram