User-controlled data in arithmetic expression

Name: User-controlled data in arithmetic expression

Description: Arithmetic operations on user-controlled data that is not validated can cause overflows.

ID: cpp/tainted-arithmetic

Kind: problem

Severity: warning

Precision: low

Performing calculations on user-controlled data can result in integer overflows unless the input is validated.

If the user is free to enter very large numbers, even arithmetic operations that would usually result in a small change in magnitude may result in overflows.

Recommendation

Always guard against overflow in arithmetic operations on user-controlled data by doing one of the following:

- Validate the user input.
- Define a guard on the arithmetic expression, so that the operation is performed only if the result can be known to be less than, or equal to, the maximum value for the type, for example `INT_MAX`.
- Use a wider type, so that larger input values do not cause overflow.

Example

In this example, a value is read from standard input into an `int`. Because the value is a user-controlled value, it could be extremely large. Performing arithmetic operations on this value could therefore cause an overflow. To avoid this happening, the example shows how to perform a check before performing a multiplication.

```
1  int main(int argc, char** argv) {
2      char buffer[20];
3      fgets(buffer, 20, stdin);
4
5      int num = atoi(buffer);
6      // BAD: may overflow if input is very large
7      int scaled = num + 1000;
8
9      // ...
10
11     int num2 = atoi(buffer);
12     int scaled2;
13     // GOOD: use a guard to prevent overflow
14     if (num2 < INT_MAX-1000)
15         scaled2 = num2 + 1000;
16     else
17         scaled2 = INT_MAX;
18 }
```

References

- Common Weakness Enumeration: CWE-190.
- Common Weakness Enumeration: CWE-191.