

Infinite loop with unsatisfiable exit condition

Name: Infinite loop with unsatisfiable exit condition

Description: A loop with an unsatisfiable exit condition could prevent the program from terminating, making it vulnerable to a denial of service attack.

ID: cpp/infinite-loop-with-unsatisfiable-exit-condition

Kind: problem

Severity: warning

Loops can contain multiple exit conditions, either directly in the loop condition or as guards around `break` or `return` statements. If none of the exit conditions can ever be satisfied, then the loop will never terminate. A program containing an infinite loop could be vulnerable to a denial of service attack if it is possible for an attacker to trigger the loop.

Recommendation

When writing a loop that is intended to terminate, make sure that all the necessary exit conditions can be satisfied and that loop termination is clear.

Example

The following example shows an infinite loop. The value of variable `i` is always zero, so the condition guarding the `break` is always false.

```
1 void test(int n) {
2     int i = 0;
3     if (n <= 0) {
4         return;
5     }
6     while (1) {
7         // BAD: condition is never true, so loop will not terminate.
8         if (i == n) {
9             break;
10        }
11    }
12 }
```

The error has been fixed below by incrementing `i` in the body of the loop.

```
1 void test(int n) {
2     int i = 0;
3     if (n <= 0) {
4         return;
5     }
6     while (1) {
7         // GOOD: condition is true after n iterations.
8         if (i == n) {
9             break;
10        }
11        i++;
12    }
13 }
```

References

- Common Weakness Enumeration: [CWE-835](#).