Static array access may cause overflow

Name: Static array access may cause overflow

Description: Exceeding the size of a static array during write or access operations may result in a buffer overflow.

ID: cpp/static-buffer-overflow

Kind: problem

Severity: warning

Precision: medium

When you use static arrays you must ensure that you do not exceed the size of the array during write and access operations. If an operation attempts to write to or access an element that is outside the range of the array then this results in a buffer overflow. Buffer overflows can lead to anything from a segmentation fault to a security vulnerability.

Recommendation

Check the offsets and sizes used in the highlighted operations to ensure that a buffer overflow will not occur.

Example

```
1  #define SIZE 30
2
3  int f(char * s) {
4      char buf[20]; //buf not set to use SIZE macro
5
6      strncpy(buf, s, SIZE); //wrong: copy may exceed size of buf
7
8      for (int i = 0; i < SIZE; i++) { //wrong: upper limit that is higher than array size
9          cout << array[i];
10     }
11 }
```

References

- I. Gerg. *An Overview and Example of the Buffer-Overflow Exploit*. IANewsletter vol 7 no 4. 2005.
- M. Donaldson. *Inside the Buffer Overflow Attack: Mechanism, Method & Prevention*. SANS Institute InfoSec Reading Room. 2002.
- Common Weakness Enumeration: CWE-119.
- Common Weakness Enumeration: CWE-131.