Too few arguments to formatting function

Name: Too few arguments to formatting function

Description: Calling a printf-like function with too few arguments can be a source of security issues.

ID: cpp/wrong-number-format-arguments

Kind: problem

Severity: error

Precision: high

Each call to the `printf` function, or a related function, should include the number of arguments defined by the format. Passing the function more arguments than required is harmless (although it may be indicative of other defects). However, passing the function fewer arguments than are defined by the format can be a security vulnerability since the function will process the next item on the stack as the missing arguments.

This might lead to an information leak if a sensitive value from the stack is printed. It might cause a crash if a value on the stack is interpreted as a pointer and leads to accessing unmapped memory. Finally, it may lead to a follow-on vulnerability if an attacker can use this problem to cause the output string to be too long or have unexpected contents.

Recommendation

Review the format and arguments expected by the highlighted function calls. Update either the format or the arguments so that the expected number of arguments are passed to the function.

Example

```
1 int main() {
2   printf("%d, %s\n", 42); // Will crash or print garbage
3   return 0;
4 }
```

References

- CERT C Coding Standard: FIO30-C. Exclude user input from format strings.
- cplusplus.com: C++ Functions.
- Microsoft C Runtime Library Reference: printf, wprintf.
- Common Weakness Enumeration: CWE-685.