C static code analysis: Appropriate size arguments should be passed to "strncat" and "strlcpy"

1 minute

Size argument of `strncat`, `strlcat` and `strlcpy` should define the size of the destination to prevent buffer overflow.

Moreover, `strncat` always adds a terminating null character at the end of the appended characters so the size argument should be smaller than the size of the destination to let enough space for it.

Noncompliant Code Example

```
void f(char* src) {
  char dest[10];
  strlcpy(dest, src, sizeof(src)); // Noncompliant; size argument is the size of the source instead of the size of the destination

  strncat(dest, src, sizeof(src)); // Noncompliant; size of the source instead of the size of the destination
  strncat(dest, src, sizeof(dest)); // Noncompliant; size argument is too large
}
```

Compliant Solution

```
void f(char* src) {
  char dest[10];
  strncat(dest, src, sizeof(dest) - 1); // Compliant
}
```