| 1 void CWE122\_Heap\_Based\_Buffer\_Overflow\_\_c\_CWE805\_wchar\_t\_memmove\_01\_bad() 2 { 3 wchar\_t \* data; 4 data = NULL; 5 /\* FLAW: Allocate and point data to a small buffer that is smaller than the large buffer used in the sinks \*/ 6 data = (wchar\_t \*)malloc(50\*sizeof(wchar\_t)); 7 **if** (data == NULL) {exit(-1);} 8 data[0] = L'\0'; /\* null terminate \*/ 9 { 10 wchar\_t source[100]; 11 wmemset(source, L'C', 100-1); /\* fill with L'C's \*/ 12 source[100-1] = L'\0'; /\* null terminate \*/ 13 /\* POTENTIAL FLAW: Possible buffer overflow if source is larger than data \*/ 14 memmove(data, source, 100\*sizeof(wchar\_t)); 15 data[100-1] = L'\0'; /\* Ensure the destination buffer is null terminated \*/ 16 printWLine(data); 17 free(data); 18 } 19 } |
| --- |

From the above code, we can see a heap buffer overflow caused by the statement “memmove(data, source, 100\*sizeof(wchar\_t))” at line 14. Specifically,“data” is a buffer with 50 “wchar\_t” elements, allocated at line 6. However, the statement at line 14 tries to move 100 “wchar\_t” elements from “source” to “data”. Thus, the buffer “data” will be overflowed and a heap overflow happens.