| 1 void CWE457\_Use\_of\_Uninitialized\_Variable\_\_struct\_array\_alloca\_no\_init\_07\_bad() 2 { 3 twoIntsStruct \* data; 4 data = (twoIntsStruct \*)ALLOCA(10\*sizeof(twoIntsStruct)); 5 **if**(staticFive==5) 6 { 7 /\* POTENTIAL FLAW: Don't initialize data \*/ 8 ; /\* empty statement needed for some flow variants \*/ 9 } 10 **if**(staticFive==5) 11 { 12 /\* POTENTIAL FLAW: Use data without initializing it \*/ 13 { 14 int i; 15 for(i=0; i<10; i++) 16 { 17 printIntLine(data[i].intOne); 18 printIntLine(data[i].intTwo); 19 } 20 } 21 } 22 } |
| --- |

From the above code, we can see the use of uninitialized variables caused by the statements at line 17 and line 18. Specifically, the code first declared a variable “data” at line 3 and make it point to a piece of memory on the stack at line 4. However, the code never initializes the data in that piece of memory before using the data at line 17 and line 18 (the code of function “printIntLine” is included in “Include/io.c”). Thus, use of uninitialized variables happens.