|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | manual.txt | report.txt | microedit | src/code.c | src/code.h |
| alex | RW | RW\* | RW\* | RW\* | W\* |
| benn | R | RW+ | RW | RW | W+\* |
| cloe |  | RW | R | R | RW+ |

RDN foo/bar

RDN foo/baz

-DN foo/cux

-DN foo/dir1

RDN foo/dir2

--- foo/dir2/flop

RDN foo/dir2/wibble

R-- foo/dir2/wobble

cp -p foo/bar foo/bar\_

rm -f foo/bar

mv foo/bar\_ foo/bar

* 1. Change owner, change group.

*pos* could be negative.

The table could be interfered with in-between calls.

int insert\_in\_table(int \*table, int val, int pos) {

**if** (pos < 0 || pos > **sizeof**(table) / **sizeof**(int)) **return** -1;

table[pos] = val;

**return** 0;

}

1. Stack canaries are (usually random) pieces of data placed in between the return address and the local variables on the stack. When a return instruction is reached, the system first checks that the stack canary is intact. This means that for a buffer overflow to successfully overwrite the return address, it has to be able to find out the value of the stack canary and make sure to only clobber it with its original value.

Dynamic bounds checking (either built-in to the programming language or specified by the programmer) checks at every point where a buffer is being indexed, that the index is within the bounds of the buffer.