Jan Rehwaldt
University of Tartu, Secure Programming Techniques, May 2012

Find all potential vulnerabilities in this C function:

```
1. int get_user_input(char *prompt)
2. {
3.         char buf[80];
4.         unsigned char input[100];
5.         int done = 0;
6.         int i, val;
7.
8.         do {
9.                 sprintf(buf, "%s> ", prompt);
10.                 printf(buf);
11.                 gets(input);
12.                 if (strlen(input) > 99) {
13.                         printf("Input too long\n");
14.                         return 0;
15.                 }
16.                 val = atoi(input);
17.                 if (val > 0) {
18.                         struct in_addr *addr;
19.                         char *buf2 = malloc(val*sizeof(addr));
20.                         for (i = 0; i < val; i++) {
21.                           if (read(0, buf2, sizeof(*addr)) < 0)
22.                                   return 0;
23.                         }
24.                         done = 1;
25.                 }
26.         } while (!done);
27.
28.         return val;
29. }
```

l. 11: gets doesn't get the size as input, which may lead to a buffer overflow

l. 12: Input size check is too late, an overflow may already have happened in l. 11

l. 16: atoi(…) input is expected to be signed char, but is unsigned (also earlier: strlen and gets)

l. 16: buffer with length 100 may lead to integer overflow

l. 19: allocated memory is never released

l. 21: not compilable as sizeof(*addr) should be sizeof(addr)

l. 21: the buffer is always overwritten and not appended as probably expected; use pread instead