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. do {
8. sprintf(buf, "%s> ", prompt);
9. printf(buf);
10. gets(input);
11. if (strlen(input) > 99) {
12. printf("Input too long\n");
13. return 0;
14. }
15. val = atoi(input);
16. if (val > 0) {
17. struct in\_addr \*addr;
18. char \*buf2 = malloc(val\*sizeof(addr));
19. for (i = 0; i < val; i++) {
20. if (read(0, buf2, sizeof(\*addr)) < 0)
21. return 0;
22. }
23. done = 1;
24. }
25. } while (!done);
26. return val;
27. }

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