Quiz: For C Programmers

15 Questions, Answers follow the “About” page.

Author: Jedidiah R. Crandall, crandaj@erau.edu

This Document was Funded by the National Science Foundation
Federal Cyber Service Scholarship For Service Program:
Grant No. 0113627

Distributed July 2002

Embry-Riddle Aeronautical University • Prescott, Arizona • USA

1. If you declare an array as A[100] in C and you try to write data to A[555], what will happen?

A. Nothing
B. The C compiler will give you an error and won't compile
C. There will always be a runtime error
D. Whatever is at A[555] will be overwritten

2. Which kinds of operations are most likely to lead to buffer overflows in C?

A. Floating point addition
B. Indexing of arrays
C. Dereferencing a pointer
D. Pointer arithmetic

3. Where can an attacker who is trying to “smash the stack” put their attack code if the buffer to be overflowed is on the stack?

A. On the stack before the return pointer
B. On the stack after the return pointer
C. In the stack frame of another function
D. On the heap
E. In a global variable

4. What can be overwritten by a buffer overflow that causes a security problem.

A. Security-sensitive data
B. A return pointer
C. Any kind of pointer
D. Anything that will make the program crash

5. What is likely to happen if you find a buffer overflow during testing by entering a random, long string for a C program?

A. The program gives you a “Buffer overflow at line X” error
B. Data is corrupted
C. The program crashes
D. The C fairy sprinkles magic memory dust on the memory that was overwritten and makes everything okay again.

6. Which of these kinds of inputs can cause a buffer overflow.

A. An environment variable
B. String input from the user
C. A single integer
D. A floating point number
E. File input

7. Which of these processes is likely to catch a buffer overflow?

A. Compilation
B. Code inspection
C. Testing by a software developer
D. Testing (or using) by a customer
E. Testing (or probing) by an attacker

8. Which of these library functions are safe as long as you tell it the correct buffer size?

A. sprintf()
B. strcpy()
C. fscanf()
D. gets()
E. memcpy()

9. Which of these is the best tool for finding unsafe library function calls?

A. The warning messages of the C compiler
B. Taping a hard-copy of the code to the wall and throwing darts at it
C. A debugger
D. A static analyzer such as ITS4

10. Which of these kinds of buffer overflows can be a security threat?

A. Stack smashing
B. Unsafe library function calls
C. Off-by-on errors where only one byte is overwritten
D. Buffer overflows in buffers that store internal data and not user input

11. If you want to use scanf() to read into a 64-byte buffer called MyBuff, which of these are correct?

A. scanf(“%s”, MyBuff);
B. scanf(“%s”, &MyBuff);
C. scanf(“%63s”, MyBuff);
D. scanf(“%64s”, MyBuff);
E. scanf(“%65s”, MyBuff);

12. Which of these attack techniques is most appropriate for a UNICODE buffer overflow?

A. Stack smashing
B. Heap imploding
C. Buffer doubling
D. The Venetian exploit

13. Which of these assumptions is always okay to make about old code used in a new project?

A. If it was already black-box tested then it doesn't need to be tested again
B. If it was already white-box tested then it doesn't need to be tested again
C. If the old code was already inspected then it doesn't need to be inspected again
D. If it limits the number of characters passed to it for every input then there will be no buffer overflows
E. None of the above

14. Which of these software engineering techniques can catch buffer overflow errors that the others might not catch?

A. Testing
B. Code inspection
C. Static analysis tools
D. Multi-platform testing

15. What can happen if a buffer overflow causes a program to crash?

A. A core dump gives the attacker access to security-sensitive data

B. A denial-of-service attack where other users on the network can no longer access that service

C. The computer can catch on fire

D. Nothing bad can happen unless the attacker is able to hijack the machine or overwrite security-sensitive data

About this Project

1. This presentation is part of a larger package of materials on buffer overflow vulnerabilities, defenses, and software practices.
   For more information, go to: http://nsfsecurity.pr.erau.edu

2. Also available are:
   - Demonstrations of how buffer overflows occur (Java applets)
   - PowerPoint lecture-style presentations on an introduction to buffer overflows, preventing buffer overflows (for C programmers), and a case study of Code Red
   - Checklists and Points to Remember for C Programmers
   - An interactive module and quiz set with alternative paths for journalists/analysts and IT managers as well as programmers and testers
   - A scavenger hunt on implications of the buffer overflow vulnerability

3. Please complete a feedback form at http://nsfsecurity.pr.erau.edu/feedback.html to tell us how you used this material and to offer suggestions for improvements.

Answers

1. D
2. B, D
3. A,B,C,D,E
4. A,B,C,D
5. B,C
6. A,B,C,D,E
7. B,C,D,E
8. A,C,E
9. D
10. A,B,C,D
11. C
12. D
13. E
14. A,B,C,D
15. A,B