




Quiz: Buffer Overflow



1. Which of these statements about the buffer overflow problem are correct?

- ☒ A. The buffer overflow problem is partly caused by the way the C language handles memory management
- ☒ B. The buffer overflow problem is partly caused by C programmers not handling their own memory management properly by checking boundaries of buffers
- ☐ C. All buffer overflows are simple programmer errors that are easily spotted
- ☒ D. Because of the complexity of the problem, buffer overflows may be overlooked by the most seasoned programmers



2. What can make a buffer overflow a security problem?

- A. Only when the attacker is able to hijack the execution of the program
- B. Only when the buffer overflow is between two computers on a network
- ☒ C. When security-sensitive data is overwritten
- ☒ D. When data that is critical to the execution of the program is overwritten causing the program to crash



3. What typically happens when a buffer is overflowed?

- A. The memory space that comes after the buffer holds the extra data as well as keeping the data that it contained before
- ☒ B. Whatever is in the memory space that comes after the buffer is overwritten
- C. The memory chip in the computer gets too big and explodes
- D. Electrons fall out of the memory chip and start a fire




4. What does it mean for a program to jump?

- A. The whole program gets moved from one place in memory to another
- ☒ B. The program starts executing instructions at a different location in memory instead of moving on to the next one
- C. The program jumps outside the bounds of a buffer to store data
- D. Jump is only used to refer to when a program starts executing instructions somewhere where it's not supposed to




5. What does a typical C program usually use stacks for?

- ☒ A. Temporary storage of variables
- B. For storing the computer-level instructions of a subroutine while the subroutine is being executed
- ☒ C. Keeping track of where it was within subroutines that called other subroutines so it knows where to resume
- D. For preventing buffer overflows



6. What prevents a typical computer from jumping and starting to execute data instead of instructions?

- ☒ A. Nothing
- B. Buffers are used to separate instructions and data
- C. The computer can always tell the difference based on what is in the memory space
- D. Data and instructions are stored in separate memory spaces on most modern computers



7. 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




8. 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




9. 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? (5)

- 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
- ALL OF THE ABOVE



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

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



11. What is likely to happen if you find a buffer overflow during testing by entering a random, long string for a C program? (4)

- 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.




12. Which of these kinds of inputs can cause a buffer overflow. (5)

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



13. Which of these processes is likely to catch a buffer overflow? (5)

- 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
- All except A.



14. 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()`



15. 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



16. Which of these kinds of buffer overflows can be a security threat? (4)


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

All of the above



17. 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



18. Which of these software engineering techniques can catch buffer overflow errors that the others might not catch? (4)

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

All of the above



19. 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