# Software Security Lab 1

Strings

# Why Strings have vulnerabilities?

Graphics- and Web-based applications, for example, make **extensive use of text input fields**, and because of **standards like XML**, **data exchanged** between programs is increasingly in string form as well. As a result, weaknesses in string representation, string management, and string manipulation have led to a broad range of software vulnerabilities and exploits.

# Standards

- In C, size_t is an unsigned integer type used to represent the size of objects in bytes. It is used to ensure that the size of any object can be represented, which is why it is the return type of the sizeof operator. Since it represents sizes, it cannot be negative.
- One of the problems with arrays is determining the number of elements. In some cases, sizeof(array) / sizeof(array[0]) to determine the number of elements in the array. However, array is a pointer type if it is a parameter. As a result, sizeof(array) is equal to sizeof(int *). For example, on an architecture (such as x86-32) where sizeof(int) == 4 and sizeof(int *) == 4, the expression sizeof(array) / sizeof(array[0]) evaluates to 1, regardless of the length of the array passed, leaving the rest of the array unaffected. So do not apply the sizeof operator to a pointer when taking the size of an array.

# Continued

- A character string literal is a sequence of zero or more characters enclosed in double quotes, as in "xyz". Array variables are often initialized by a string literal and declared with an explicit bound that matches the number of characters in the string literal. A better approach is to not specify the bound of a string initialized with a string literal because the compiler will automatically allocate sufficient space for the entire string literal, including the terminating null character.
- Improperly bounded string copies occur when data is copied from a source to a fixed-length character array (for example, when reading from standard input into a fixed-length buffer).
- Strings must contain a null-termination character at or before the address of the last element of the array before they can be safely passed as arguments to standard string-handling functions, such as strcpy() or strlen().

# Continued

- The strncpy function in C is used to copy a specified number of characters from one string to another. It's particularly useful when you want to avoid buffer overflows by specifying the exact number of characters to copy, including the null terminator if it's encountered before the limit is reached.