The Chinese University of Hong Kong

Department of Information Engineering

Summer Workshop – Fun with Information Engineering and Security

Lab 6 – Web Security 2 and System Security

Introduction

In this lab, we will dive into another topic of web security, including the concepts of databases and SQL. We will then explore security attacks related to the database, and the mitigation of these attacks. Besides, we will also go through some system security concepts on the boundary check and buffer overflow. As a wrap-up, reading materials on Linux and CTF are also provided.

Section 1: Web Security 2

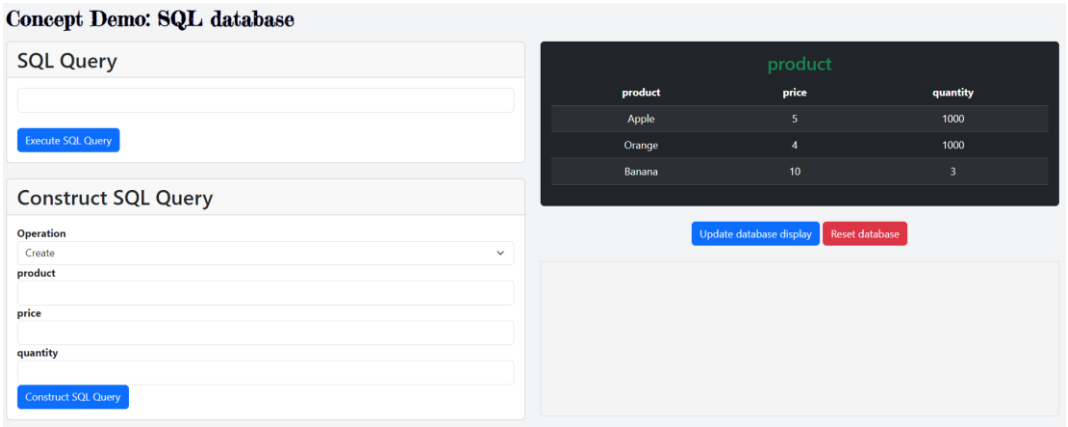
Section 1.1: Database and SQL

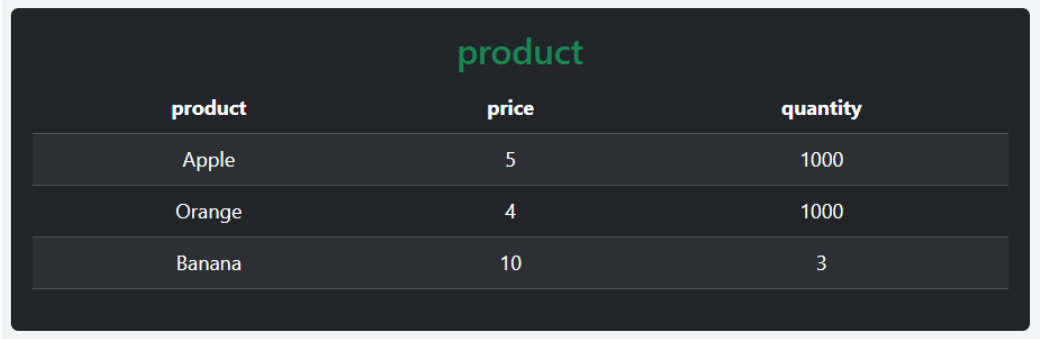
A database is an organized collection of structured data that is stored and managed electronically. It is designed to store, retrieve, and manage large amounts of data in a structured and consistent manner. Databases are widely used in various applications such as web applications.

There are different types of databases, with the most common ones being relational databases (e.g., MySQL, Oracle, PostgreSQL) and non-relational databases (e.g., MongoDB, Redis). Relational databases use Structured Query Language (SQL) to store, manage and retrieve data.

Try it!

Go to https://iesummerworkshop.github.io/sql.html





This is what we have:

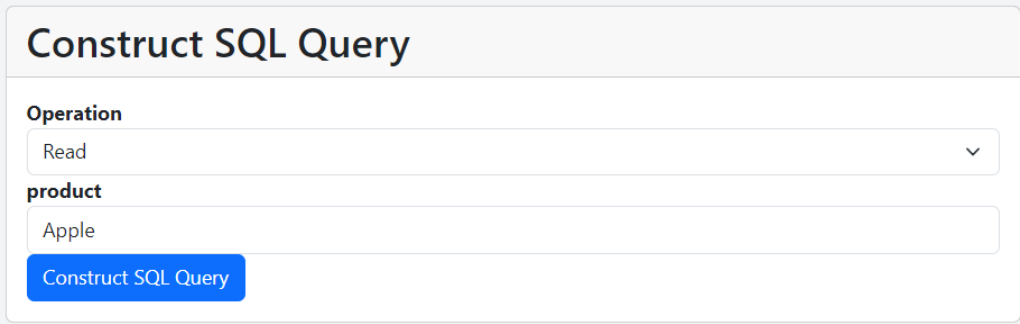
- A **table:** `product`
- Three **columns**: `product`, `price`, `quantity`
- Three **records**, with the first record of: `Apple`, `5`, `1000`

In real life, we need to construct SQL queries to interact with the table.

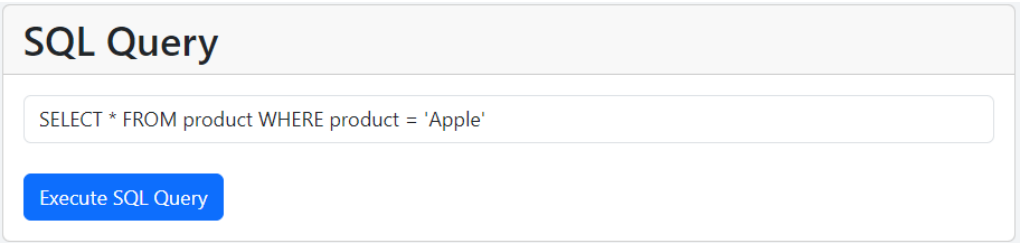
Reading a value from a table

Under the "Construct SQL Query" section:

- Select "Read" as the operation
- Input "Apple" in the product field

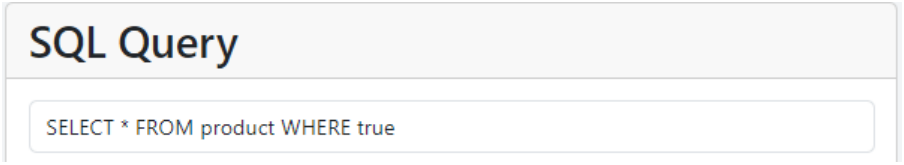


Click on Construct SQL Query button, you should see a SQL query generated in the SQL Query section:



Select all fields from the product table and filter records with the product name 'Apple'. Click on Execute SQL Query

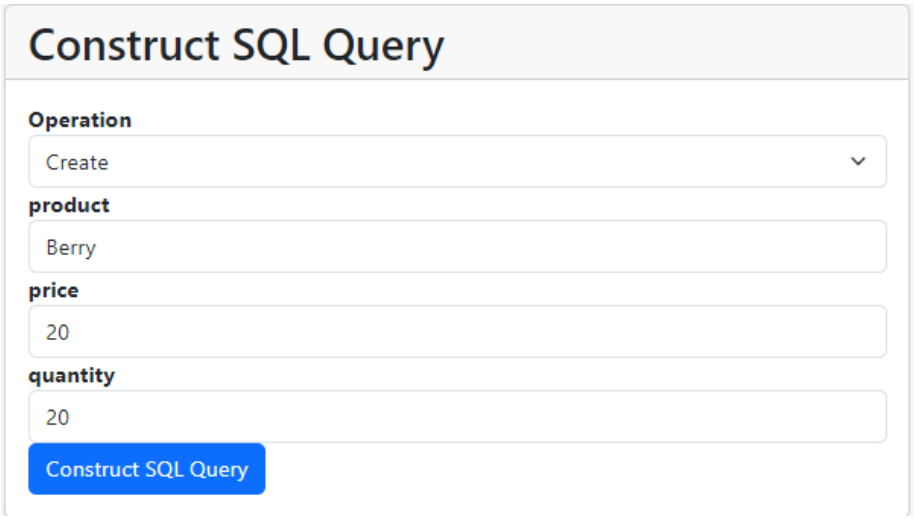
1. What is the output?
2. How to obtain the price of Orange?
3. What is the output if you search for a product that does not exist in the table?
4. What is the output of `SELECT * FROM product WHERE true`? Paste the SQL query in the query section



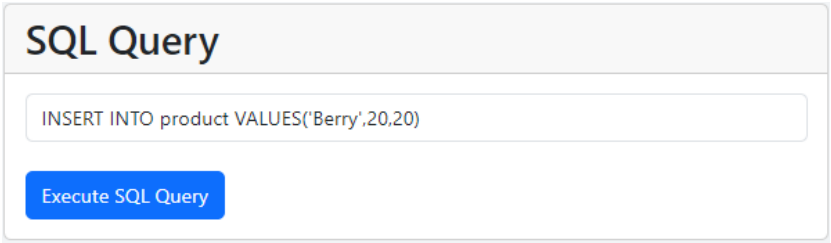
Create a record from a table

Under the "Construct SQL Query" section:

- Select "Create" as the operation
- Input value for the product, price, and quantity field



Click on Construct SQL Query button, you should see a SQL query generated in the SQL Query section:



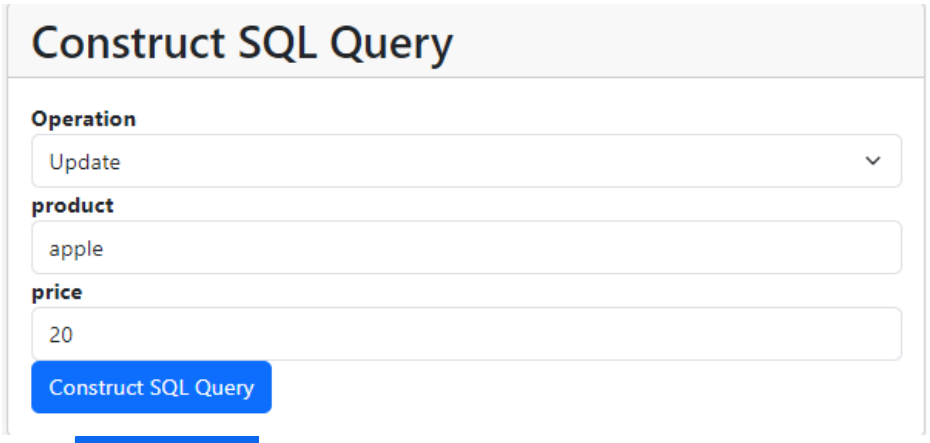
As we are adding values for all the columns of the table, we don't need to specify the column names in the SQL query. However, the value must be in the same order as the columns in the table. Click on Execute SQL Query

1. What is the output? What are the changes on the product table?

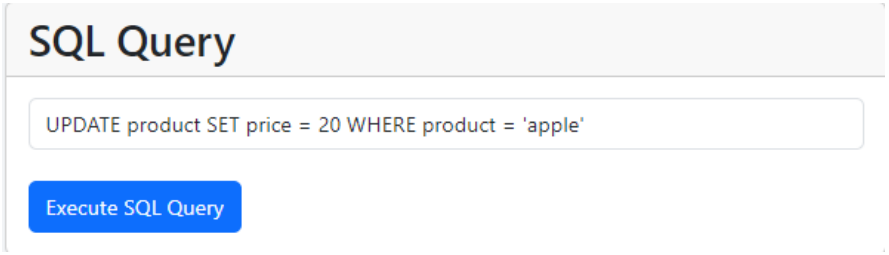
Update value from a table

Under the "Construct SQL Query" section:

- Select "Update" as the operation
- Input value for the product and price



Click on Construct SQL Query button, you should see a SQL query generated in the SQL Query section:



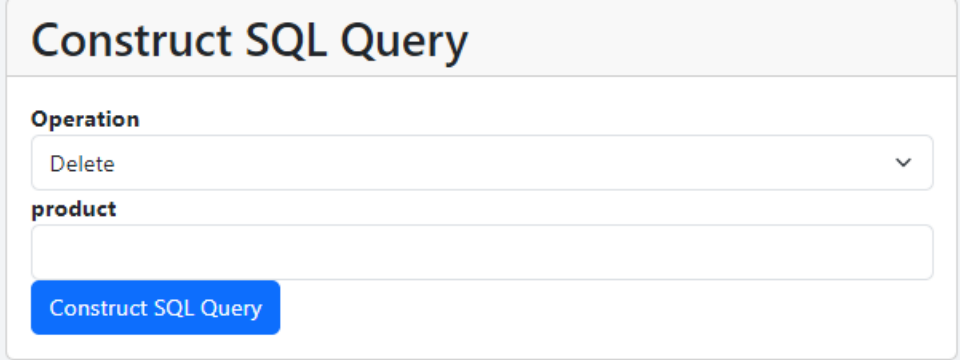
The update statement is used to modify the existing records in the table. Click on Execute SQL Query

1. What is the output? What are the changes on the product table?
2. How to change the quantity of apples from 1000 to 100?
3. How to add 5 to the price when the product has a quantity of 1000?
4. What is the output if the product does not exist?

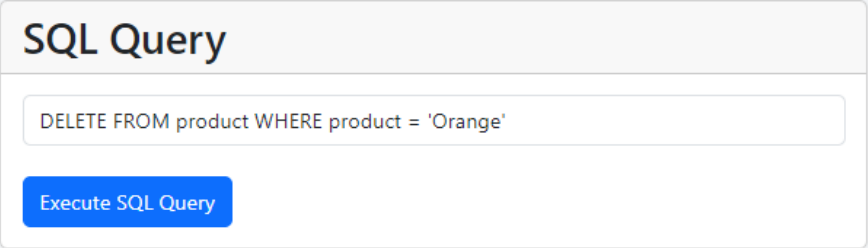
Delete record from a table

Under the "Construct SQL Query" section:

- Select "Delete" as the operation
- Input value for the product field



Click on Construct SQL Query button, you should see a SQL query generated in the SQL Query section:



The delete statement is used to delete the existing records in the table. Click on Execute SQL Query

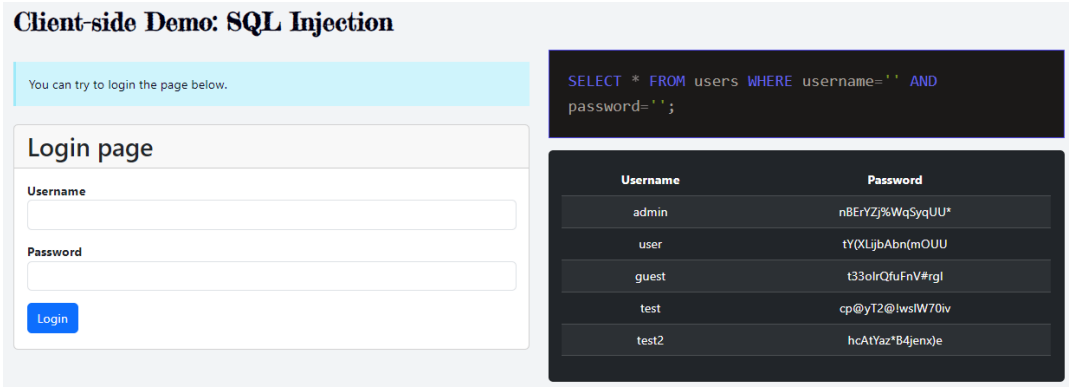
1. What is the output? What are the changes on the product table?
2. What is the output if the product does not exist?

Section 1.2: SQL Injection

SQL Injection is a security vulnerability that occurs when an attacker can manipulate an SQL query by inserting malicious SQL code into a query string. This can lead to unauthorized access, data breaches, or data manipulation. Similar to XSS, it is one of the OWASP top 10 web application security risks. SQL Injection attacks happen when a web application does not properly validate or sanitize user input before using it in an SQL query.

Try it!

Go to https://iesummerworkshop.github.io/sql-injection.html

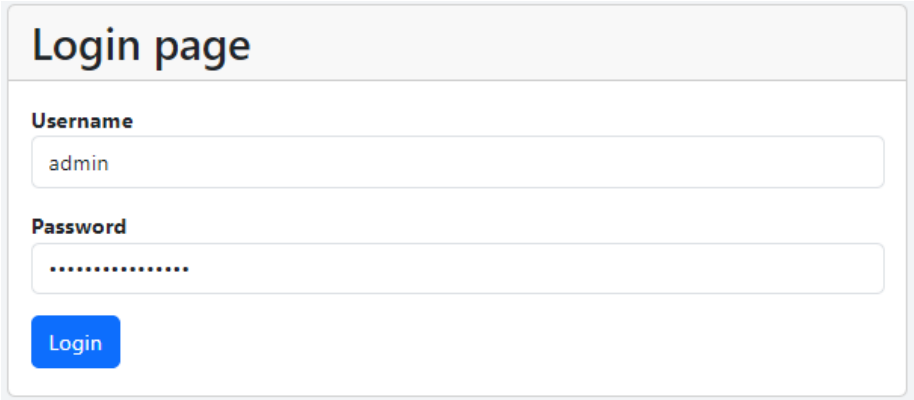


This page is a login page connected with a user table, which contains the username and password. The SQL query is given:

```
SELECT * FROM users WHERE username='' AND password='';
```

The query is searching for a user record that username = ??? and password = ??? at the same time. If such a user exists, that means the user inputs a correct username and password and the server would allow the user login.

Let's try to log in with the admin account. In the user table, given that the username is admin and the password is nBErYZj%WqSyqUU*. Input the value in the login form:

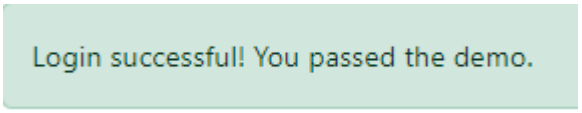


The SQL query is updated correspondingly:

```
SELECT * FROM users WHERE username='admin' AND
password='nBErYZj%WqSyqUU*';
```

Click on the Login button, you should get the following message upon successful

authentication:



However, in real-life applications, the database lies on the backend. End users cannot interact with the database directly. How can we login to the system without knowing the username and password?

From lab 1, we learn about the truth table:

- AND operation: both statements are true, for the result to be true
- OR operation: either statement is true, for the result to be true

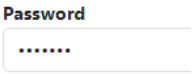
From the previous section, we already know the result of `SELECT * FROM product WHERE true`

By combining the above, we can craft a malicious query in the login form:

1. The username and password form interprets the input as text (quote inside `''`) how can we get rid of the quote symbols so that the OR operator gives the semantic meaning?
2. The AND and OR operator works from left to right. Should the attack string placed in the username and password field?
3. Is there any way to remove the AND operator?

Sample solution 1:

Put the following string into the password field: `'or''='`



The query should look like the following:

```
SELECT * FROM users WHERE username='' AND password=''or''='';
```

- The first `'` get rid of the quotation, so the operator OR will not be considered as a string
- Similar to the above, the last `'` get rid of the quotation
- Craft a statement that the output will be true `''=''`. Any string comparison with true as the result also works, e.g. `'1'='1'`

Stmt1	Stmt2	Stmt1 AND Stmt2 (tempStmt)	Stmt3	tempStmt OR Stmt3
`username=''` → false	`password=''` → false	false	`''=''` → true	true

Sample solution 2:

Put the following string into the password field: `'or true--`

The query should look like the following:

```
SELECT * FROM users WHERE username=''or true--' AND password='';
```

- The first ‘ get rid of the quotation, so the operator OR will not be considered as a string
- Craft a statement that the output will be true. Any comparison with true as the result also works, e.g. `1=1`
- `--` is the comment annotation in SQL query. By applying `--`, the command after it will not be interpreted

Stmt1	Stmt2	Stmt1 OR Stmt2
`username=’’` → false	true	true

Section 1.3: Mitigation of SQL Injection

The following are some common strategies to defend against SQL Injection:

1. Use Parameterized Queries (Prepared Statements). Utilize prepared statements provided by your programming language or framework. These mechanisms separate SQL code from user input, ensuring that input is treated as data rather than executable code.
2. Input Validation and Sanitization: Validate and sanitize user input before using it in database queries. Apply strict input validation to ensure that only expected data types and formats are accepted.
3. Web Application Firewall (WAF): Employ a web application firewall to detect and block SQL injection attempts. A WAF can analyze incoming requests, identify suspicious patterns, and block or alert on potential SQL injection attacks.

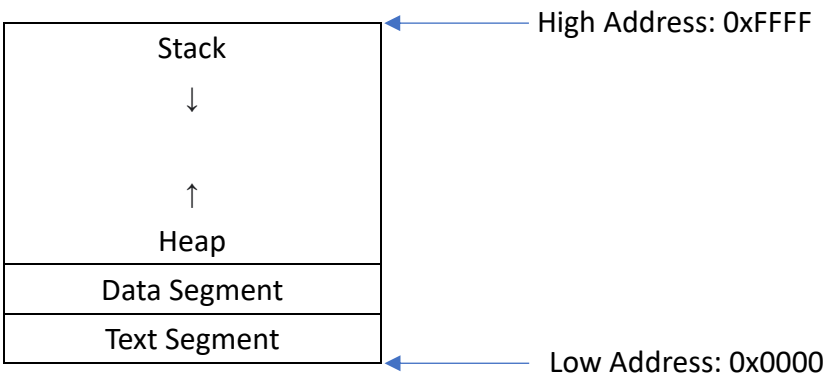
Section 2: System security

Section 2.1: Memory layout

A memory representation of a program consists of the following sections:

- Text Segment
  - Begin at the lower address
  - Stores the program code
- Data segment
  - Stores the global, static variables and constants
- Heap
  - Dynamic storage spaces, e.g. allocated by `malloc()` in c language
- Stack
  - Begin at the higher address
  - Stores local variables, function arguments

* The heap and stack share the same space, where they are growing toward each other



Section 2.2: Boundary check on programming language

Boundary check is a validation process of verifying that any access to array elements or other data structures is within the valid range of indices. This is crucial for preventing errors such as reading from or writing to memory locations that are outside the bounds of the array, which can lead to undefined behavior, crashes, or security vulnerabilities.

Among hundreds of programming languages, most of the language handles boundary checks automatically, while some of them don't. The most famous language that does not handle boundary checks is C/C++.

Try it!

Compare how Python and C handle boundary checks. The following are the same set of code written in Python and C respectively:

- Initialize an array `my_array` with value `[1, 2, 3]`
- Use a for loop to loop over the list, intentionally reading out of bound

Python

Go to https://iesummerworkshop.github.io/py-lab.html

Copy the below code snippet into the coding section

```python
my_array = [1, 2, 3]
upper_limit = 5

for i in range(0, upper_limit):
    print("my_array[{}] = {}".format(i, my_array[i]))
```

What is the output?

C

Go to https://www.onlinegdb.com/online_c_compiler

Copy the below code snippet into the coding section

```c
#include <stdio.h>

int main() {
    int my_array[3] = {1, 2, 3};
    int upper_limit = 5;

    for (int i = 0; i < upper_limit; i++) {
        printf("my_array[%d] = %d\n", i, my_array[i]);
    }
    return 0;
}
```

What is the output?

Python performs automatic boundary checks and raises an IndexError for out-of-bounds access; while C does not perform boundary checks, leading to undefined behavior if array bounds are violated (probably segmentation fault / unknown value). It is entirely the user's responsibility to handle the bounds of the array.

Section 2.3: Buffer overflow

Buffer overflow is a type of vulnerability that occurs when a program writes more data to an array than it can hold. This can lead to unexpected behavior, including crashes, data corruption, and security vulnerabilities such as arbitrary code execution.

Try it!

With previous hands-on practice, we know that the following code snippets are unsafe and intentionally going out of bounds:

```c
#include <stdio.h>

int main() {
    int my_array[3] = {1, 2, 3};
    int upper_limit = 5;

    for (int i = 0; i < upper_limit; i++) {
        printf("my_array[%d] = %d\n", i, my_array[i]);
    }
    return 0;
}
```

The following are the correct ways to handle the C code:

Go to https://www.onlinegdb.com/online_c_compiler

Copy the below code snippet into the coding section

```c
#include <stdio.h>
int main() {
    int my_array[3] = {1, 2, 3};
    int upper_limit = 5;
    int length_of_arr = sizeof(my_array) / sizeof(my_array[0]);

    if (upper_limit > length_of_arr){
        printf("Upper limit out of bound");
        return 0;
    }
    else{
        for (int i = 0; i < upper_limit; i++) {
            printf("my_array[%d] = %d\n", i, my_array[i]);
        }
    }
    return 0;
}
```

1. What is the output?
2. Why the above code snippet does not cause the error?

Some functions in the C library are considered unsafe now as these functions blindly accept parameters without performing any checking. Safe functions are introduced now and developers are encouraged to use the safe version.

Try it!

Let's try on one of a C string standard function `strcpy()`. It is a function to assign a string to an array. `strcpy()` is the unsafe version, while `strncpy()` is the safe version.

Unsafe C code

Go to https://www.onlinegdb.com/online_c_compiler

Copy the below code snippet into the coding section

```c
#include <stdio.h>
#include <string.h>

int main() {
    char arr [10];
    strcpy(arr, "Hello World");
    printf("%s", arr);

    return 0;
}
```

What is the output?

Safe C Code

Go to https://www.onlinegdb.com/online_c_compiler

Copy the below code snippet into the coding section

```c
#include <stdio.h>
#include <string.h>

int main() {
    char arr [10];
    strncpy(arr, "Hello World", 10);
    printf("%s", arr);

    return 0;
}
```

What is the output?

How does this function prevent out-of-bound memory read-write?

Section 3: Further reading

Besides the content of the three-day workshop, there are a lot more topics worth discussing in the field of information engineering and security:

Section 3.1: Linux

Linux is an open-source operating system (OS) (software acts as a middleman between computer hardware and software). Some well-known operating systems are Windows 7/8/10, MacOS, and iOS. Where another example is Android is a Linux-based operating system. Around 95% of servers run on Linux. Below are some reasons why technology professional uses Linux:

- Linux is considered more secure than other OS as the source code is open to the public. Professionals can identify vulnerabilities and deal with them promptly, so they are less susceptible to attacks.
- Linux is considered more reliable compared to other OS, with a good user experience
- Most Linux distributions are free
- Linux offers various tools and methods to identify and mitigate security risks. Some of the useful tools are encryptions, firewalls, and intrusion detection systems.

One of the most well-known Linux distributions in the cybersecurity field is Kali Linux. It is designed for forensics and penetration testing. Kali Linux pre-installed various networking and security tools.

The following are resources to get to know more about Linux:

Title	Content of the page	URL
Linux/Unix Tutorial	- Introduce basic Unix/Linux command - Basic concepts of Unix/Linux system - Package management in	https://www.geeksforgeeks.org/linux-tutorial/

	Linux - User, group permission - Networking tools	
Official website of Kali Linux	- Documentation for Kali Linux - Kali image download	https://www.kali.org/
Kali Linux Tutorial	- Guide to install Kali Linux - Tutorial on Kali penetration test and forensic tools	https://www.tutorialspoint.com/kali_linux/index.htm

Section 3.2: Capture the Flag

Capture the Flag is a kind of security game/competition in that the challengers solve a variety of tasks ranging from basic programming, web security, system security, etc. The platform usually hides a piece of information (i.e. the flag) and the participant's task is to recover or retrieve the hidden flag.

Below are some resources to start with:

Title	Content of the page	URL
CTF Resources	- Introduce the knowledge to solve the CTF puzzle	https://ctfs.github.io/resources/
Pico CTF	- Online CTF platform	https://picoctf.org/ https://play.picoctf.org/practice
CTFtime	- Information on the recent CTF competition - Writeup on CTF questions	https://ctftime.org/

Section 4: Challenge time

Now, please go to https://iesummerworkshop.github.io/rpg/page.html to enjoy your challenge! Walk in front of the computer and press enter to interact with it. Mark down the key you get!

1. [☆ ☆ ☆ ☆] Shopping center

Go to https://iesummerworkshop.github.io/shopping.html

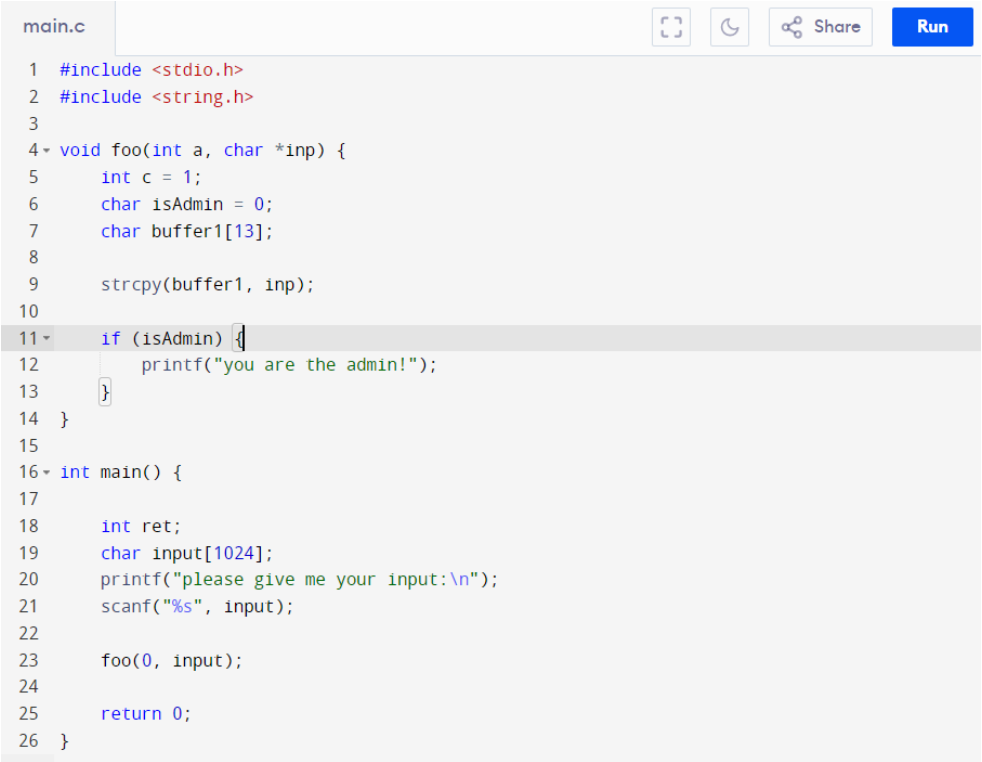
Welcome to the shopping center! Your target is to obtain the information on the flag. Click on the flag to get the key!

Hints:

A. The flag has not been released for sale yet.
B. How does the webpage know what to display?
C. Is there any input field where we can launch an attack?
D. Can you make the webpage display an error message so that it will potentially expose the SQL query?
E. Open the developer console and refresh the page. The console should provide some useful information

2. [☆ ☆ ☆] Buffer Overflow

Go to: https://www.programiz.com/online-compiler/2Cca581JYpIwj



Copy the below code snippet under “main.c” if necessary

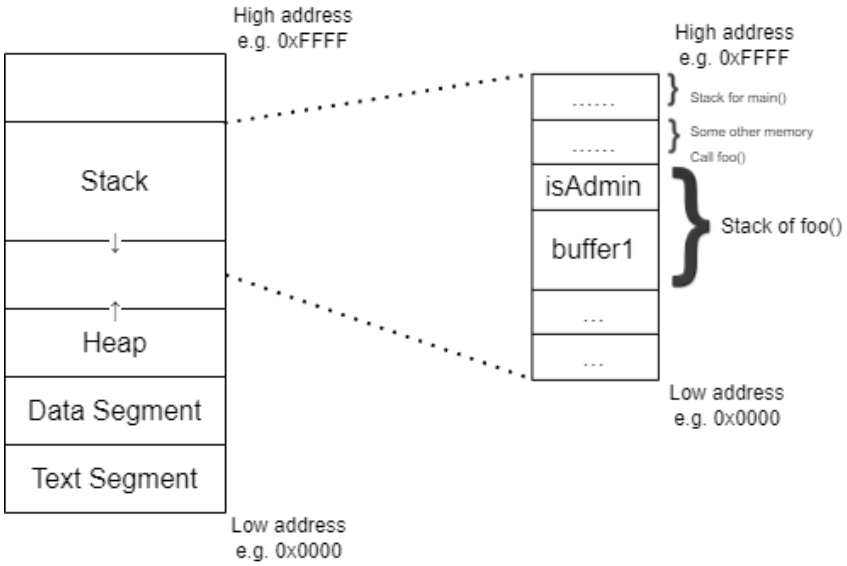
```c
#include <stdio.h>
#include <string.h>

void foo(int a, char *inp) {
    int c = 1;
    char isAdmin = 0;
    char buffer1[13];

    strcpy(buffer1, inp);

    if (isAdmin) {
        printf("you are the admin!");
    }
}

int main() {
```

```
    int ret;
    char input[1024];
    printf("please give me your input:\n");
    scanf("%s", input);

    foo(0, input);

    return 0;
}
```

Question: What should you give as input to get the announcement that `you are the admin`?

Hints:

1. This is how the memory looks when the program runs



2. Data on stack write to the memory from low address to high address, i.e. from bottom to top in the above diagram
3. In C language, it is considered True if it is a non-zero value
4. How can you overwrite the value of isAdmin?

This webpage visualizes how this C program works:

https://pythontutor.com/render.html#code=%23include%20%3Cstdio.h%3E%0A%23include%20%3Cstring.h%3E%0A%0Avoid%20foo%28int%20a,%20char%20*inp%29%20%7B%0A%20%20%20%20int%20c%20%3D%201%3B%0A%20%20%20%20char%20isAdmin%20%3D%200%3B%0A%20%20%20%20char%20buffer1%5B13%5D%3B%0A%0A%20%20%20%20strcpy%28buffer1,%20inp%29%3B%0A%0A%20%20%20%20if%20%28isAdmin%29%20%7B%0A%20%20%20%20%20%20%20%20printf%28%22you%20are%20the%20admin!%22%29%3B%0A%20%20%20%20%7D%0A%7D%0A%0Aint%20main%28%29%20%7B%0A%0A%20%20%20%20int%20ret%3B%0A%20%20%20%20char%20*input%20%3D%20%22aaaaaaaaaaaa%22%3B%0A%0A%20%20%20%20foo%280,%20input%29%3B%0A%0A%20%20%20%20return%200%3B%0A%7D&cppShowMemAddrs=true&cumulative=false&curInstr=0&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false

This link may help you understand more about buffer overflow:
https://lightfootlabs.io/resources/Learn-Buffer-Overflows-through-Visuals

Interact with this counter to submit the key for lab 5 and lab 6

Please also help us to fill the questionnaire:

https://forms.gle/teHBEnDbesSCrEJBA