### Question 1

Prompt: Explain how you solved the “Docker World” and “Key Delivery Service” challenges. You should describe the key concepts involved. For example, you could mention the netcat utility, how SSH authenticates with RSA keypairs, etc. See Project 1, Part 1 description for more details.

To solve the Docker World challenge, I used a computer networking utility called netcat. Netcat which is accessed through the terminal on macOS by using the command nc followed by arguments, is particularly useful for reading from and writing to networking connections using the TCP or UDP protocol over IPv4 or IPv6 addresses. Using the man nc command in the terminal window, it shows a list of common uses which include, shell-scrip based HTTP clients and servers, network daemon testing, TCP proxies, and much more.

By typing nc followed by the given IP address & port number, we’re prompted to say hello, which when done gives us the flag.

For the Key Delivery Service challenge , I typed the terminal command provided next to the SSH tag. After hitting enter, it prompted for a password, which I did not have so I couldn’t log in. I hit control C to stop the current task in terminal.

Then I typed the command next to the key delievery service which uses netcat. Upon hitting enter I noticed a private RSA key was provided.RSA is a type of asymmetric encryption that is widely used in many products and services. Asymmetric encrytion uses a public/private key pair which solves the problem of needing to send a key to someone along with an encrypted message. With RSA, your public key is like your mailbox, and your private key is what's used to open it, as well as sign letters of your own to send to someone.

In the case of this challenge, the private key provided by the key delivery service was enough capture the next flag. I did this by going to my hidden ssh directory on macOS and copied the private RSA key file. I pasted in the new private key and saved.

By passing in the acquired RSA key to the challenge host through our modified file, I was able to log in. Typing ls showed a flag file which when doing cat flag gave me the final flag for the challenge.

I was able to log in by passing the acquired RSA key to the challenge host through ssh. Then typing ls showed a flag file, which then typing cat flag gave the challenge flag.

### Question 2

Prompt: Explain how you solved the “SQL Injection 1” challenge. You should describe the key concepts involved. For example, you should describe how SQL injection works. See Project 1, Part 1 description for more details.

The first steps I took to solve the SQL Inection challenge was to look at the login.php source code in order to see where it could be vulnerable to the attack.

Upon inspecting , You can see this line of code here where the database query will occur after the user inputs their username and password. This is where I took advantage of, and solved the SQL Injection challenge by using a 1=1 is Always true technique.

SQL Injection normally occurs when a website asks for user input, like a username or password and instead of the expected input, the attacker types in a SQL statement that will unknowingly run on the website's database. This allows the attacker to interfere with queries in the database, gaining access to protected data.

In this case I typed in ‘ OR 1=1 ‘ --. The double lines at the end of the statement make all other lines of SQL after it appears as comments, bypassing any restrictions. When this statement is executed by the SQL database, it will return all rows from the users table since OR 1=1 is always true, which allows us to log in. This is what the code will look like behind the scenes when executed. (SHOW IMAGE)

After filling in both fields with the same input and pressing enter. I was able to log into the website successfully and capture the challenge flag.