

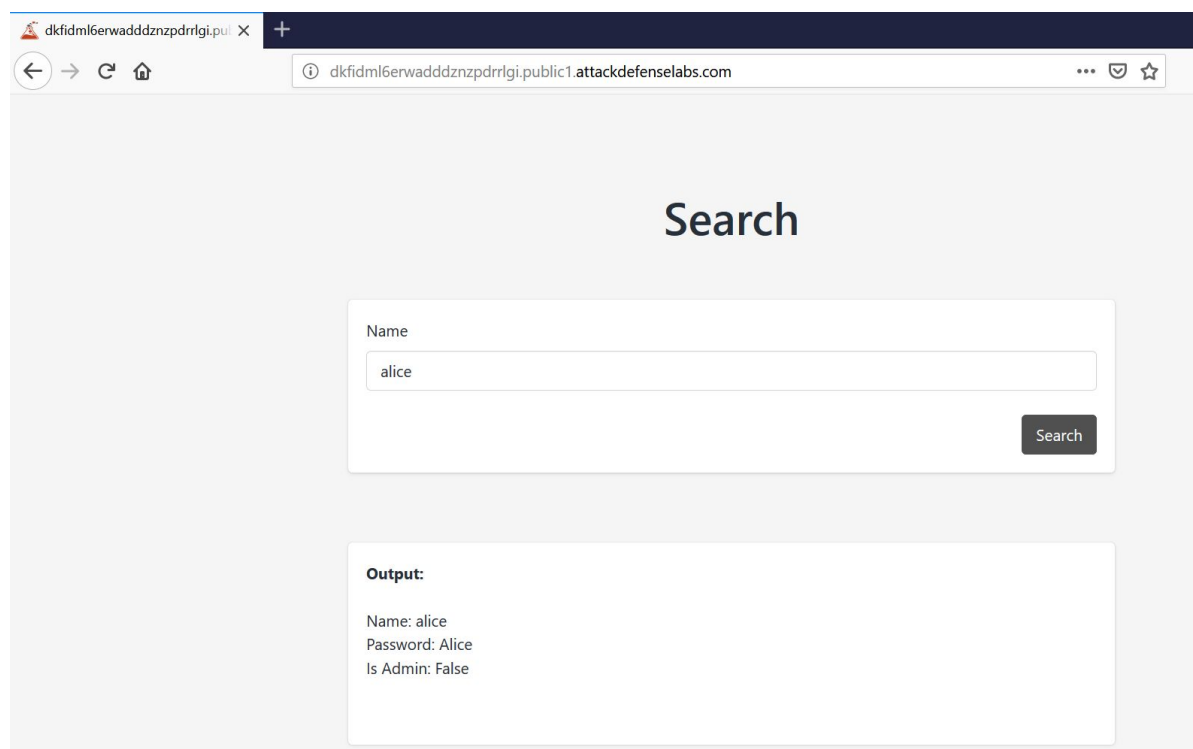
[illegible]

Name	MongoDB: NoSQL injection
URL	https://www.attackdefense.com/challengedetails?cid=232
Type	Infrastructure Attacks: MongoDB

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

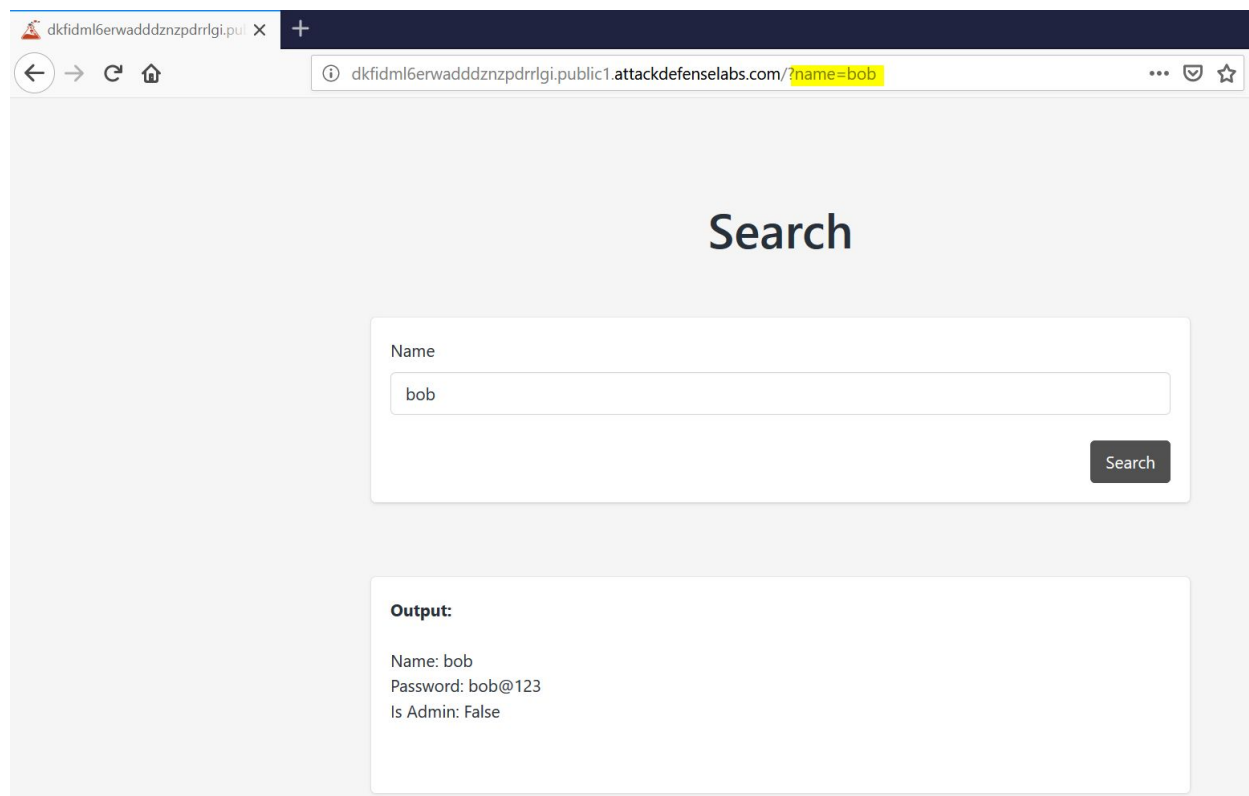
The webapp is vulnerable to injection, which can be exploited to dump all documents from the collection.

Step 1: Interact with the web application.



The screenshot shows a web browser window with the address bar displaying `dkfidml6erwadddznpdrrlgi.public1.attackdefense.com`. The page has a light gray background with the word "Search" in a large, bold, black font at the top center. Below this, there is a white rectangular box containing a search form. The form has a label "Name" above a text input field. The input field contains the text "alice". To the right of the input field is a dark gray button with the word "Search" in white. Below the search form is another white rectangular box labeled "Output:". Inside this box, the following text is displayed: "Name: alice", "Password: Alice", and "Is Admin: False".

Upon entering a random name say bob in the search field, if the record exist for that name, details are shown otherwise nothing is shown.

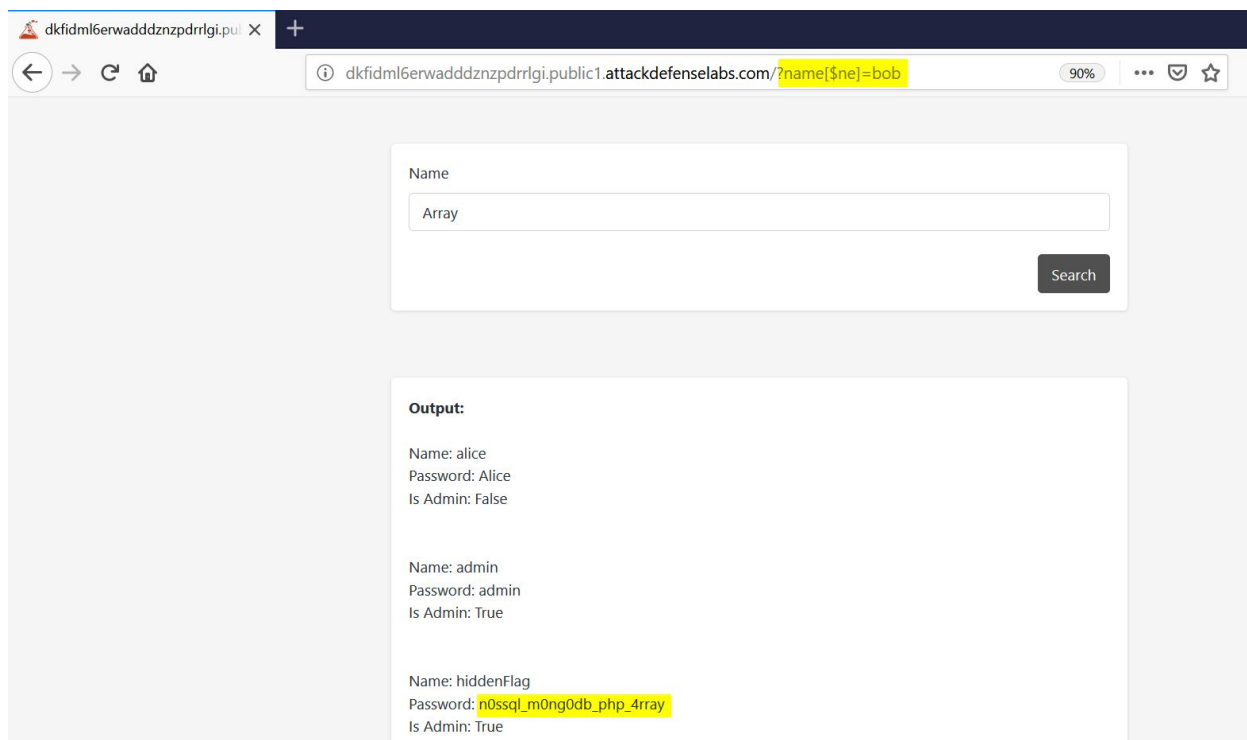


The screenshot shows a web browser window with the URL `dkfidml6erwaddddznzpdrrlgi.public1.attackdefenselabs.com/?name=bob`. The page has a light gray background and a large heading "Search" in the center. Below the heading is a search form with a label "Name" and a text input field containing "bob". To the right of the input field is a dark gray "Search" button. Below the search form is an "Output:" section containing the following text:

```
Name: bob
Password: bob@123
Is Admin: False
```

Step 2: Inject payload in the URL

Payload: `/?name[$ne]=bob`



Flag: n0ssql_m0ng0db_php_4rray

References:

1. MongoDB (<https://www.mongodb.com/>)
2. Mongodb is vulnerable to SQL injection in PHP at least (<https://www.idontplaydarts.com/2010/07/mongodb-is-vulnerable-to-sql-injection-in-php-at-least/>)